

ECOLE NATIONALE SUPERIEURE DE L'AERONAUTIQUE

CENTRE D'ETUDES ET DE RECHERCHES EN AUTOMATISME

ETUDE D'UNE METHODE DE RECHERCHE D'UN MINIMUM
DANS UN ESPACE A N DIMENSIONS SANS CONTRAINTE

+

QUELQUES CONSIDERATIONS SUR DEUX METHODES
DE MINIMISATION LE LONG D'UNE DIRECTION
DANS L'ESPACE

F. SAEZ VACAS
Groupe IV

S O M M A I R E

Introduction

Notations

I - LE PROBLEME DE LA MINIMISATION, CONSIDEREE COMME UNE
APPROXIMATION AU SECOND DEGRE

1 - Généralités

2 - Le problème de la convergence quadratique

2.1 - Interprétation géométrique

II - ALGORITHME DE DAVIDON

3 - Organigramme de fonctionnement

3.1 - Stabilité de la méthode

3.2 - Convergence quadratique

4 - Remarques sur la méthode

III - ANALYSE DE LA MINIMISATION UNIDIMENSIONNELLE DANS LE
CONTEXTE DE LA MINIMISATION A N DIMENSIONS

5 - Davidon

6 - Fibonacci

7 - Analyse en parallèle de Davidon et Fibonacci

7.1 - Nécessité de stratégies supplémentaires

7.1.1 - Une nouvelle stratégie

7.2 - Conclusions provisoires

- 8 - Fonctions unimodales, mais pas fortement unimodales
- 9 - Discussion de la contrainte d'unimodalité dans M.M.U.
 - 9.1 - Stratégie pour la M.M.U. de Davidon
 - 9.2 - Stratégie pour la M.M.U. de Fibonacci
- 10 - Choix de l'intervalle initial dans la M.M.U.
 - 10.1 - On connaît la valeur exacte ou estimée du minimum de la fonction
 - 10.2 - On ne connaît rien sur la valeur du minimum de la fonction
- 11 - Test de convergence

IV - LISTE DE PROGRAMMES ET RESULTATS

V - REMARQUES FINALES

- 12 - Bibliographie sur la méthode de Davidon et sur une autre méthode de minimisation qui ne calcule pas des dérivées
- 13 - Application de la méthode à l'optimisation paramétrique de systèmes dynamiques

VI - REFERENCES

Introduction

Les méthodes d'optimisation statiques (c'est-à-dire des systèmes dont les paramètres n'évoluent pas avec le temps) peuvent se diviser en deux grandes classes : les méthodes directes et les méthodes indirectes. Les premières localisent le vecteur optimum par des mouvements stratégiques dans l'espace correspondant. Elles nécessitent la connaissance de la valeur de la fonction critère à chaque point, mais non la forme algébrique, ni ses dérivées.

Les méthodes indirectes se basent sur des propriétés mathématiques de la fonction et en particulier des dérivées.

Il existe une abondante bibliographie dans les deux classes.

Ce rapport présente une méthode de convergence quadratique appartenant à la deuxième classe ci-dessus. Il s'agit de la méthode adaptative de Davidon (1959), modifiée par Fletcher et Powell (1963) et modifiée aussi quelque peu par nous mêmes.

On fait ici un exposé complet de la méthode, ce qui est justifié par les causes suivantes :

- a - Elle est peu connue (le rapport de Davidon est plus ou moins introuvable).
- b - Les exposés de Fletcher et Powell, étant tout à fait opposés au contenu réel de leurs articles, il est difficile de croire qu'ils en aient contribué à une vraie diffusion.

- c - A notre connaissance il n'existe en langue française aucun travail relatif à ce sujet.
- d - Il y a des points faibles que les auteurs passent sous silence.

Ainsi, par exemple, nous essayons de montrer que la clé insoupçonnée du succès de la méthode de Davidon est la méthode de minimisation unidimensionnelle (dite aussi linéaire ou à un seul paramètre) qui, de plusieurs façons contrôle l'efficacité et la convergence de celle-là.

De l'analyse des méthodes de minimisation unidimensionnelle (M.M.U.) dans le contexte des méthodes de minimisation à n dimensions (M.M.N.), on tire des conséquences diverses, parmi lesquelles il y en a qui sont générales et applicables à d'autres M.M.N. et qui peuvent (peut-être) expliquer quelques défauts signalés par les auteurs. Nous proposons des solutions diverses et des possibilités de correction concrétisées dans les M.M.U. de Davidon et Fibonacci.

Il nous semble que, plus important que les solutions apportées, est l'analyse elle-même. Pourtant à la fin on programme l'une d'entre elles, plus la méthode générale, bien sûr, à laquelle on munie d'options et des flexibilités. On a voulu un programme assez général ce qui entraîne comme conséquence logique une surabondance pour des cas particuliers.

L'utilisateur aura donc intérêt à alléger les options du programme et très probablement à faire appel à d'autres possibilités décrites au cours de l'analyse qui suit et même à prévoir d'autres modifications.

Notations

Les notations vectorielles qu'on utilise par la suite sont empruntées à la mécanique quantique. Elles sont peut être un peu lourdes mais ont l'avantage payant de ne pas constituer une source constante de confusion.

$|X\rangle$ vecteur colonne

$\langle X|$ " ligne

$\langle X|Y\rangle$ produit scalaire

$|X\rangle\langle Y|$ " vectoriel

Pour les matrices on utilise des lettres capitales, par exemple : $A = [A_{ij}]$. Ainsi, si l'on suppose satisfaites les conditions des opérations avec les matrices :

$A|x\rangle$ vecteur colonne

$\langle x|A$ " ligne

$\langle x|A|y\rangle$ scalaire

Lorsqu'il est indifférent, qu'il s'agisse d'un vecteur ligne ou colonne, on utilise la variable soulignée, par exemple $f(\underline{X})$

I - LE PROBLEME DE LA MINIMISATION, CONSIDEREE COMME UNE APPROXIMATION AU SECOND DEGRE.

1 - Généralités

Soit :

$$f(\underline{X}) = f_0 + \langle \underline{X} | \underline{b} \rangle + \frac{1}{2} \langle \underline{X} | \underline{A} | \underline{X} \rangle \quad (1)$$

la forme quadratique standard par rapport à un vecteur \underline{X} dans un espace métrique à n dimensions de norme euclidienne où :

$|\underline{X}\rangle$ est un vecteur variable ($n \times 1$)

$|\underline{b}\rangle$ est un vecteur constant ($n \times 1$)

\underline{A} une matrice constante, symétrique, définie strictement positive (c'est-à-dire dont les valeurs propres sont positives) $n \times n$

En dérivant (1), une fois, par rapport à \underline{X} :

$$|g(\underline{X})\rangle = |\underline{b}\rangle + \underline{A} |\underline{X}\rangle \quad (2)$$

qui est le gradient ($n \times 1$) en un point $|\underline{X}\rangle$, et en appliquant la condition de stationnarité $|g(\underline{X})\rangle = |0\rangle$ on trouve, puisque \underline{A} est symétrique définie strictement positive (cf Boudarel et al. [6]) que la fonction (1) à un minimum en :

$$|\underline{X}_{\min}\rangle = -\underline{A}^{-1} |\underline{b}\rangle \quad (3)$$

pour lequel :

$$|g(\underline{X}_{\min})\rangle = |0\rangle$$

Les contours de (1) sont des hypérellypsoïdes dont les axes principaux sont les vecteurs propres de A de longueurs inversement proportionnelles aux valeurs propres ; $|X_{\min}\rangle$ est leur centre de symétrie.

Propriété 1

En prenant comme point de départ un point $|X\rangle$ de (1) on atteint le minimum (3) par un déplacement égal au vecteur $-A^{-1}|g(\underline{X})\rangle$

En effet, prémultipliant (2) par A^{-1} et en tenant compte de (3)

$$\begin{aligned} A^{-1}|g(\underline{X})\rangle &= A^{-1}|b\rangle + |X\rangle = -|X_{\min}\rangle + |X\rangle \\ \Rightarrow |X_{\min}\rangle &= |X\rangle - A^{-1}|g(\underline{X})\rangle \end{aligned} \quad (4)$$

Propriété 2

En prenant comme point de départ un point $|X\rangle$ de (1) on atteint le minimum (3) par n déplacements successifs le long des n directions conjuguées par rapport à A et linéairement indépendantes. Le point de départ X_{i-1} pour le déplacement ième serait le minimum de la fonction (1) sur la direction (i-1)-ième.

En particulier les vecteurs propres sont des directions conjuguées linéairement indépendantes.

Les méthodes qui font usage des deux propriétés que l'on vient d'énoncer sont dites de convergence quadratique.

2 - Le problème de la convergence quadratique.

Considérons maintenant le cas général où $f(\underline{X})$ est quelconque et possède un minimum défini. Au voisinage dudit minimum $f(\underline{X})$ est en général suffisamment décrite par une approximation au deuxième degré tel que l'équation (1) ; dans ce cas \underline{b} et A sont les matrices des dérivées premières et secondes de la fonction.

Dans la mesure où nous sommes dans un point suffisamment proche du minimum, la propriété (4) nous permet de localiser ce dernier d'un seul coup. Pourtant la plupart du temps ce n'est pas le cas et il résulte "antiéconomique" d'évaluer la matrice A^{-1} (inverse de celle des dérivées secondes) dans des points éloignés de l'optimum. Et ce qui est pire, très souvent est contradictoire, puisque ne s'agissant pas en fait d'une hyperquadrique, l'application de (4) peut nous éloigner davantage.

Bref, une méthode algorithmique de minimisation sera efficace si elle est conçue de façon à converger tout le temps, et que, arrivée au voisinage du minimum, elle s'adapte à la géographie au second degré pour converger désormais avec rapidité quadratique. La méthode de Davidon appartient à cette classe.

2.1 - Interprétation géométrique

(On suppose $f(\underline{X})$ définie et différentiable partout). \underline{X} représente les coordonnées d'un point dans un espace linéaire à n dimensions. L'ensemble de \underline{X} pour lequel $f(\underline{X})$ est constante, forme une surface $(n - 1)$ dimensions appelée surface de réponse ou surface de niveau. Il s'agit d'une relation

d'équivalence, que divise l'ensemble des $\{X\}$ en deux sous ensembles disjoints. Or, il existe autant de relations d'équivalence que l'ensemble des valeurs possibles de $f(\underline{X})$ a d'éléments, lequel en général, est un sous-ensemble de la droite réelle. Donc, ordonné et avec un élément premier ou minorant.

Lorsqu'on établit la relation d'équivalence correspondante à cet élément on trouve normalement que le sous-ensemble équivalent des deux dans lesquels est divisé l'espace $\{X\}$ a un seul élément. Si c'est le cas, on dit que la fonction possède un point minimum bien défini.

Chaque $\{X\}$ fait partie d'une des surfaces de cette famille et le gradient de la fonction caractérise cette dernière en ce point.

$$|g(\underline{X})\rangle = \left| \frac{\partial f(\underline{X})}{\partial \underline{X}} \right\rangle \quad (5)$$

A son tour $|g(\underline{X})\rangle$ peut être considéré comme un point dans un autre espace à n dimensions, où l'on définit de la même façon des relations d'équivalence par rapport au domaine de $f(\underline{X})$ (figure 1).

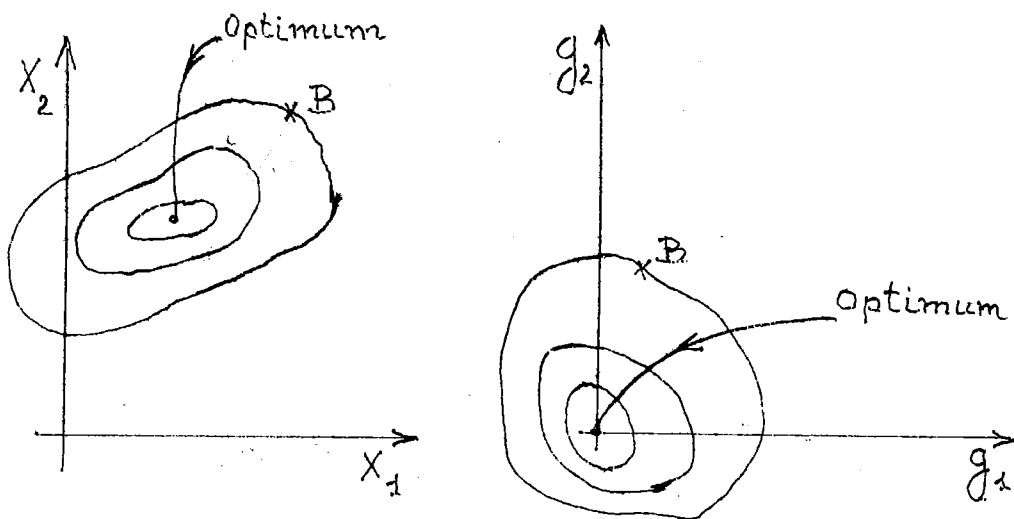


Figure 1

Au voisinage de tout point $|B\rangle$ les dérivées secondes de $f(\underline{X})$ définissent une application linéaire des variations de position en des variations de gradient, selon :

$$X|dg(\underline{X})\rangle = A |dX\rangle \quad (6)$$

ou.

$$A_{ij} = \frac{\partial^2 f(\underline{X})}{\partial X_i \partial X_j} \quad (7)$$

Les vecteurs $|dg(\underline{X})\rangle$ et $|dX\rangle$ ont la même direction seulement lorsque $|dX\rangle$ est un vecteur propre de A.

On peut faire deux hypothèses :

a - A est constante

b - A représente avec une approximation suffisante l'application des termes de degré supérieur à 1 dans le développement de Taylor de $f(\underline{X})$

$|X\rangle$ proche à $|X_{\min}\rangle$

Dans ces cas, il suffirait, comme l'on a déjà dit, de connaître le gradient $|g\rangle$ en un point pour déterminer l'emplacement du minimum ; la variation désirée sur $|g\rangle$ serait $-|g\rangle$:

$$-|g(\underline{X})\rangle = A |\Delta X\rangle \quad (8)$$

d'où :

$$|\Delta X\rangle = -A^{-1} |g(\underline{X})\rangle \quad (9) \equiv (4)$$

nous exprime la variation qu'il faudrait imprimer à $|X\rangle$.

La méthode de Davidon suppose que l'on ne connaît pas si l'on est dans un de ces deux cas et élabore itérativement une succession de matrices $\{S_0, S_1, S_2 \dots S_k \dots\}$ symétriques et strictement définies positives. Si A est constante on démontre que, au bout de $N \leq n$ itérations (n étant la dimension de l'espace) la méthode situe le point minimum et $\{S\} \rightarrow A^{-1}$. Si A dépend de $|X\rangle$ on démontre que la méthode converge toujours vers le point minimum et, S_0 pouvant être indifféremment quelconque, on arrive à un point où l'hypothèse b) commence à être vraie, à partir duquel on atteint le minimum en $N \leq n$ itérations.

Avant de décrire l'algorithme de l'élaboration de la matrice S , voyons sa signification physique.

Si dans l'équation (9) on remplace A^{-1} par S , S représente une application linéaire des variations de $|g(\underline{X})\rangle$ dans des variations de $|X\rangle$. Nous allons démontrer que, si S est effectivement A^{-1} , $\frac{1}{2} \langle g(\underline{X}) | S | g(\underline{X}) \rangle$ représente une mesure de la quantité par laquelle $f(\underline{X})$ excède sa valeur minimale :

On déduit de (8) :

$$- \langle \Delta X | g(\underline{X}) \rangle = \langle \Delta X | A | \Delta X \rangle \quad (10)$$

et remplaçant $|\Delta X\rangle$ à gauche de (10) par sa valeur (9), transposant convenablement ($\langle \Delta X = - \langle g(A^{-1})$).

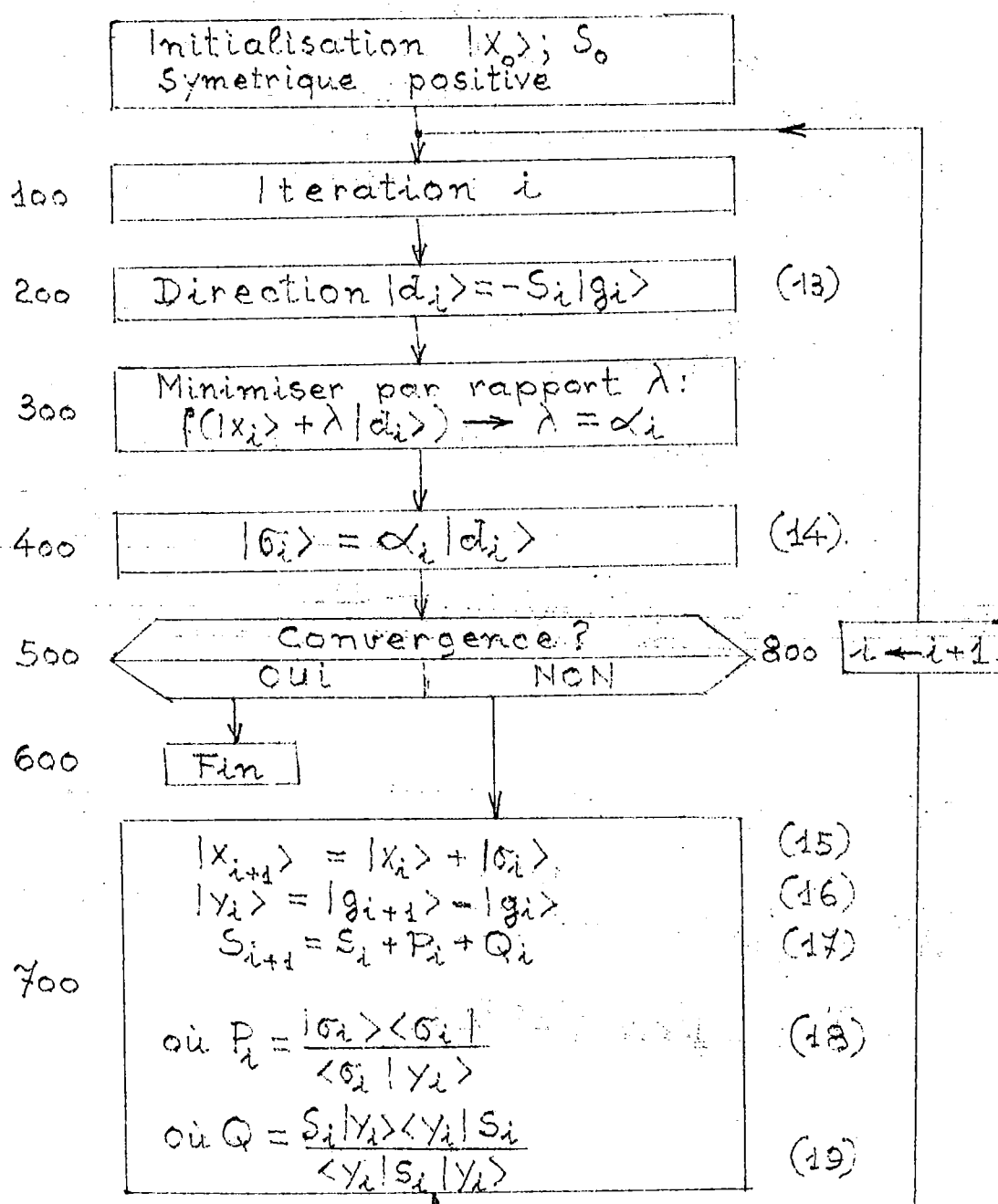
$$\frac{1}{2} \langle g | A^{-1} | g \rangle = \frac{1}{2} \langle \Delta X | A | \Delta X \rangle \quad (11)$$

$$\frac{1}{2} \langle g | S | g \rangle = \frac{1}{2} \langle \Delta X | A | \Delta X \rangle$$

II - ALGORITHME DE DAVIDON

3 - Organigramme de fonctionnement

La méthode a été décrite par W.C. Davidon [1] et modifiée plus tard par Fletcher et Powell [2]. C'est sous cette seconde forme que nous la présentons ici.



Les chiffres de gauche sont associés à des étiquettes du programme de l'algorithme. Les chiffres de droite sont des repères d'équations.

S_0 doit être de préférence une estimation de A^{-1} , si possible, ou une autre quelconque (I par exemple).

On suppose satisfaite par $|\sigma_i\rangle$ la condition de tangence à une surface de niveau ou, ce qui revient au même, d'orthogonalité avec le gradient :

$$\langle \sigma_i | g_{i+1} \rangle = 0 \quad (20)$$

condition associée au bloc 300.

Il faut démontrer que :

3.1 - L'algorithme est stable ($f(\underline{X}_{i+1}) \leq f(\underline{X}_i)$), donc la méthode est valable pour n'importe quelle fonction définie et différentiable.

3.2 - Dans le cas d'une hyperquadrique ;

$f(|X_0\rangle + |\sigma_0\rangle + |\sigma_1\rangle + \dots + |\sigma_k\rangle + \dots + |\sigma_N\rangle)$ est la valeur minimum de la fonction ($N \leq n$)

3.1 - Stabilité de la méthode

Le vecteur direction $|d_i\rangle$ donné par équation (13) est dirigé toujours du côté où la fonction diminue, ce qui formellement s'exprime :

$$-\langle d_i | g_i \rangle = \langle g_i | S_i | g_i \rangle \quad (21)$$

scalaire positif pour toute valeur de $|g_i\rangle$, c'est-à-dire pour tout $|X_i\rangle$. Le scalaire $-\langle d_i | g_i \rangle = -\langle g_i | d_i \rangle$ représente la valeur avec signe contraire de la dérivée de la fonction sur la direction $|d_i\rangle$.

Démonstration de 3.1 :

Elle consiste à prouver que S_i est strictement définie positive ($\rightarrow \alpha_i$ positif). Puisque S_0 est imposée strictement positive, l'argument récurrent le suppose vrai pour S_i , démontre qu'il l'est aussi pour S_{i+1} , donc il l'est pour tout i .

Recurrence :

(Artifice : définir $|p\rangle = (S_i)^{1/2} |X\rangle$ et $|q\rangle = (S_i)^{1/2} |Y_i\rangle$ et appliquer l'inégalité de Schwartz à ces deux vecteurs :
 norme $(|p\rangle) \times$ norme $(|q\rangle) \geq |\langle p | q \rangle|$)

$$\begin{aligned} \langle X | S_{i+1} | X \rangle &\stackrel{(17), (18), (19)}{=} \langle X | S_i | X \rangle + \frac{\langle X | \sigma_i \rangle \langle \sigma_i | X \rangle}{\langle \sigma_i | Y_i \rangle} - \\ &\quad - \frac{\langle X | S_i | Y_i \rangle \langle Y_i | S_i | X \rangle}{\langle Y_i | S_i | Y_i \rangle} = \frac{\langle p | p \rangle \langle q | q \rangle - \langle p | q \rangle^2}{\langle q | q \rangle} + \\ &\quad + \frac{\langle X | \sigma_i \rangle^2}{\langle \sigma_i | Y_i \rangle} \geq \frac{\langle X | \sigma_i \rangle^2}{\langle \sigma_i | Y_i \rangle} \end{aligned}$$

mais

$$\begin{aligned} \langle \sigma_i | Y_i \rangle &\stackrel{(16)}{\rightarrow} \langle \sigma_i | g_{i+1} \rangle - \langle \sigma_i | g_i \rangle \stackrel{(20)}{\rightarrow} -\langle \sigma_i | g_i \rangle \\ &\stackrel{(13), (14)}{\rightarrow} \alpha_i \langle g_i | S_i | g_i \rangle \stackrel{(21)}{\rightarrow} > 0 \end{aligned} \quad (21')$$

Donc

$$\langle X | S_{i+1} | X \rangle \geq \frac{\langle X | \sigma_i \rangle^2}{\alpha_i \langle g_i | S_i | g_i \rangle} \geq 0 \quad \text{pour tout } |X\rangle \quad (22)$$

Fin de la recurrence

Fin de la démonstration 3.1

Remarque importante

Il est à signaler que la satisfaction stricte de la condition (20) est fondamentale dans la démonstration précédente. En particulier, si ce n'était pas ainsi, on n'aurait pas le droit de dire que la méthode est toujours stable. En avançant les événements, disons que l'approximation avec laquelle (20) est satisfaite, est liée à la précision de la minimisation unidimensionnelle réalisée par le bloc 300 de l'organigramme et le danger pour que (22) ne soit pas vrai croît au fur et à mesure que $|X\rangle$ s'approche du minimum ($\langle g_i | S_i | g_i \rangle$ est petit avec une grande probabilité).

Il convient de remarquer en plus que, même si l'on réussit dans la recherche à avoir un point tel que $f(\underline{X}_{i+1}) \leq f(\underline{X}_i)$, ceci n'assure pas toujours que S_{i+1} est strictement définitive positive. Or, si S_{i+1} n'est pas strictement définitive positive il n'est pas possible, de savoir a priori où peut nous amener la méthode (cf section 9.1).

3.2 - Convergence quadratique

On suppose A constante.

On va montrer que $|\sigma_0\rangle, |\sigma_1\rangle, |\sigma_2\rangle, \dots, |\sigma_k\rangle, \dots$ sont les vecteurs propres, linéairement indépendants de $S_{K+1}A$, dont la valeur propre est l'unité. D'où l'on déduit que $S_n A$ est la matrice unité ($S_n = A^{-1}$).

Démonstration de 3.2

$$\begin{aligned} |Y_i\rangle &\xrightarrow{(16)} |g_{i+1}\rangle - |g_i\rangle \\ &\xrightarrow{(2)} A|X_{i+1}\rangle - A|X_i\rangle \\ &\xrightarrow{(15)} A|\sigma_i\rangle \end{aligned} \quad (23)$$

Formons $S_{i+1} A |\sigma_i\rangle$:

$$\begin{aligned} S_{i+1} A |\sigma_i\rangle &\xrightarrow{(23)} S_{i+1} |Y_i\rangle \xrightarrow{(17), (18), (19)} S_i |Y_i\rangle + \frac{|\sigma_i\rangle \langle \sigma_i|}{\langle \sigma_i | Y_i \rangle} |Y_i\rangle - \\ &- \frac{S_i |Y_i\rangle \langle Y_i | S_i | Y_i \rangle}{\langle Y_i | S_i | Y_i \rangle} |Y_i\rangle = |\sigma_i\rangle \end{aligned} \quad (24)$$

(24) nous dit que $|\sigma_i\rangle$ est un vecteur propre de $S_{i+1}A$, dont la valeur propre est l'unité. Dans (25) et (26) nous avons le premier pas de la récurrence :

Recurrence : $S_K A |\sigma_i\rangle = |\sigma_i\rangle$

$$\text{pour } 0 \leq i < K \quad (25)$$

est valable pour $K = 1$, selon (24)

$$\langle \sigma_i | A | \sigma_j \rangle = 0$$

$$\text{pour } 0 \leq i < j < K \quad (26)$$

(26) est vraie pour $j = 1$:

$$\langle \sigma_0 | A | \sigma_1 \rangle = 0$$

puisque

$$\begin{aligned} \langle g, | \sigma_0 \rangle &\xrightarrow{S_1 (20)} 0 \\ &\xrightarrow{(24)} \langle g, | S_1 A | \sigma_0 \rangle \\ &\xrightarrow{(13), (14)} -\frac{1}{\alpha_1} \langle \sigma_1 | A | \sigma_0 \rangle \end{aligned}$$

On suppose (25) et (26) pour K quelconque.

$$|g_K\rangle \xrightarrow{(2)} |b\rangle + A |X_K\rangle$$

$$\xrightarrow{(15)} |b\rangle + A (|X_{i+1}\rangle + |\sigma_{i+1}\rangle + |\sigma_{i+2}\rangle + \dots + |\sigma_{K-1}\rangle) =$$

$$= |g_{i+1}\rangle + A (|\sigma_{i+1}\rangle + |\sigma_{i+2}\rangle + \dots + |\sigma_{K-1}\rangle) \quad (27)$$

$$\langle \sigma_i | g_k \rangle \xrightarrow{(26), (27), S_i (20)} \langle \sigma_i | g_{i+1} \rangle = 0$$

$$\text{pour } 0 \leq i < K \quad (28)$$

multipliant (25) scalairement par $|g_k\rangle$:

$$\langle g_k | S_k A | \sigma_i \rangle \xrightarrow{(25)} \langle g_k | \sigma_i \rangle$$

$$\xrightarrow{(13), (14)} -\frac{1}{\alpha_k} \langle \sigma_k | A | \sigma_i \rangle$$

$$\xrightarrow{(28)} 0$$

$$\longrightarrow \langle \sigma_k | A | \sigma_i \rangle = 0 \quad 0 \leq i < K \quad (29)$$

(29) prouve (26) pour tout K.

multipliant (25) scalairement par $|y_k\rangle$:

$$\langle y_k | S_k A | \sigma_i \rangle \xrightarrow{(25)} \langle y_k | \sigma_i \rangle \xrightarrow{(23)} \langle \sigma_k | A | \sigma_i \rangle$$

$$\xrightarrow{(29)} 0 \quad 0 \leq i < K \quad (30)$$

Pour terminer l'induction, il faut démontrer que

$$S_{K-1} A |\sigma_i\rangle = |\sigma_i\rangle \quad 0 \leq i < K$$

$$S_{K+1} A |\sigma_i\rangle \xrightarrow{(17), (18), (19), (29), (30)} S_K A |\sigma_i\rangle$$

$$\xrightarrow{(25)} |\sigma_i\rangle \quad 0 \leq i < K \quad (31)$$

Fin de la recurrence

L'équation (31) prouve que $|\sigma_0\rangle, |\sigma_1\rangle, |\sigma_2\rangle, \dots, |\sigma_{n-1}\rangle$ sont les vecteurs propres de $S_n A$; (29) en plus ils sont conjugués par rapport à A , donc linéairement indépendants \rightarrow

$$\rightarrow S_n A = I \rightarrow S_n = A^{-1}$$

Le minimum est situé par conséquent après n iterations.

Fin de la démonstration 3.2

4 - Remarques sur la méthode

La matrice S est formée symétrique, strictement définie positive. Sa mission consiste à produire une transformation de la variation du gradient (équation (9)) dans la variation de $|X\rangle$ de façon à minimiser (12). Elle doit conserver l'information accumulée, en ajoutant celle recueillie au cours de l'itération, sur la base des changements obtenus dans $|X\rangle$ et $|g\rangle$, et s'adapter quadratiquement près du minimum.

$\{P_i\}$ est telle que

$$A^{-1} = \sum_{i=0}^{n-1} P_i \quad (32)$$

L'expression de Q_i est construite de façon à assurer la satisfaction de (24).

(Pour plus de détails sur la formation de P_i et Q_i cf [2])

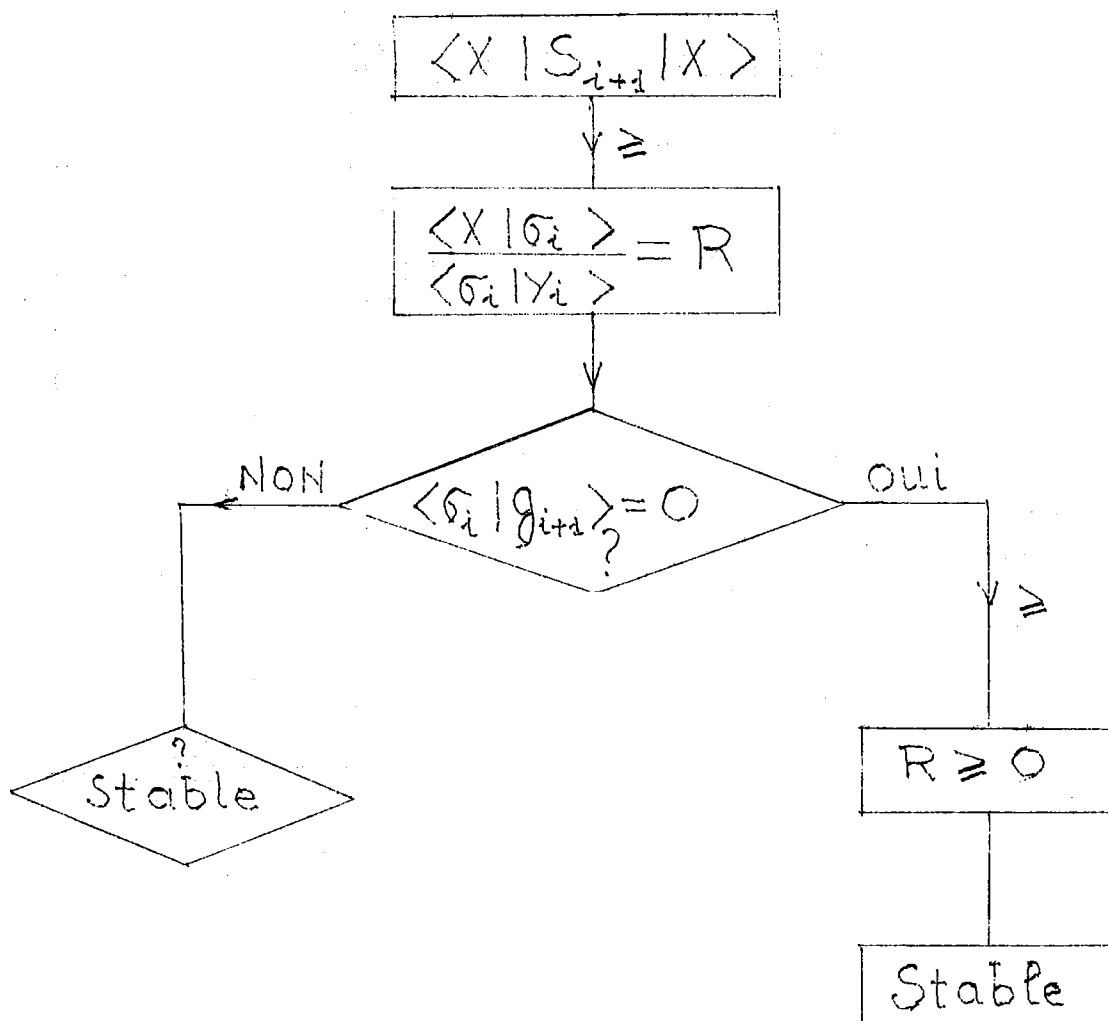
En résumé, la méthode de Davidon est une méthode de minimisation dans un espace à n dimensions sans contrainte, générale, adaptative, continuellement convergente et dont la dynamique de convergence est quadratique dans la mesure où sont satisfaites les hypothèses a) et b) de la section 2.1. Elle présente un intérêt supplémentaire dans le sens qu'elle produit une matrice S qui contient toute l'information descriptive de la géographie au 2ème degré du système autour de l'optimum, ce qui est important vis à vis de la sensibilité de ce dernier par rapport aux paramètres.

Ici s'achève tout ce qu'on peut dire sur l'algorithme en soi, qui est facilement programmable dans un langage symbolique (Fortran, Algol).

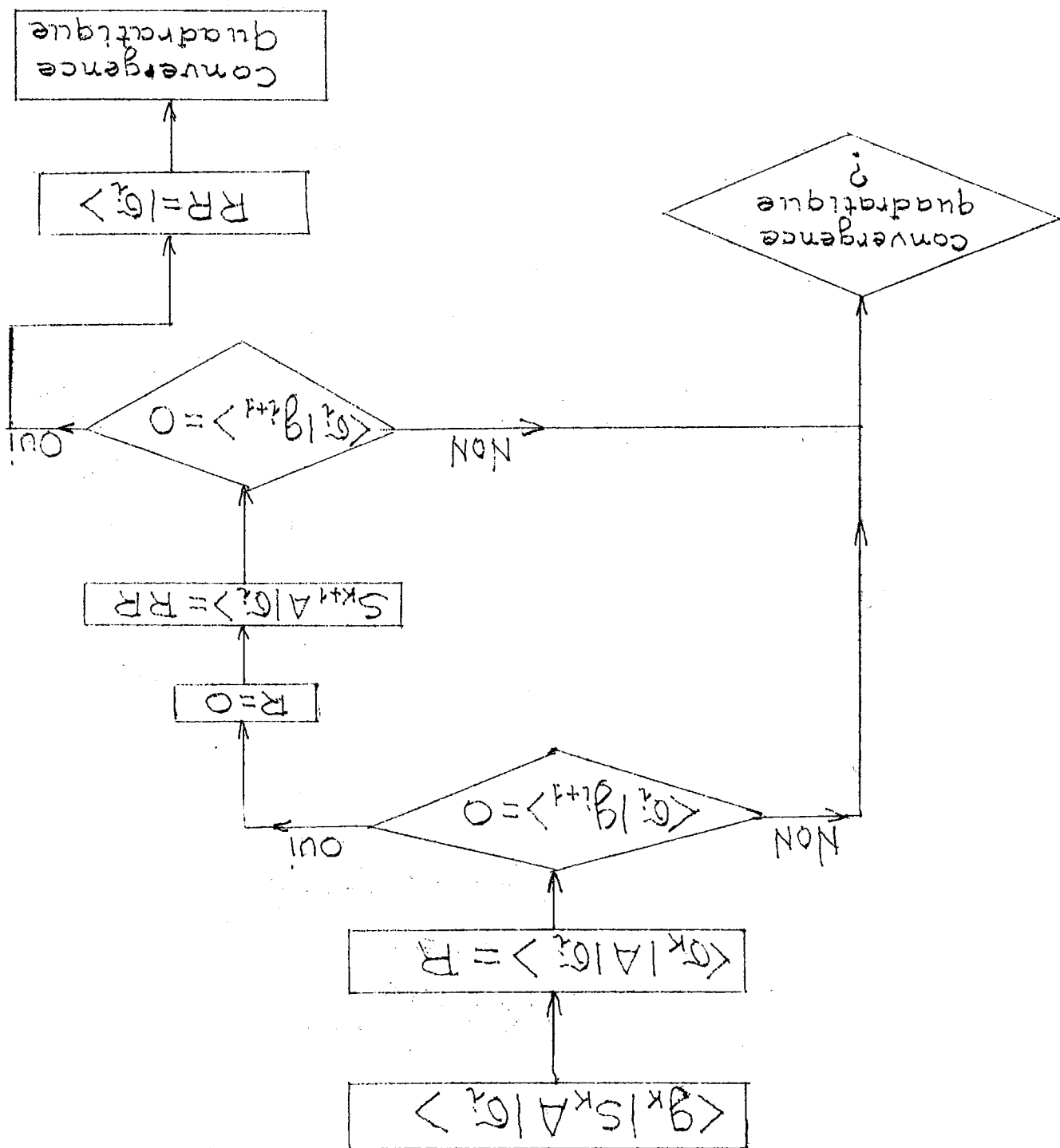
Il s'agit simplement d'une transcription des formules de l'organigramme qu'on a déjà vu, sauf 4.1 le sous-programme de minimisation représentée par le bloc 300 et 4.2 le test de convergence (bloc 500).

4.1 - La réalisation du programme de minimisation pose un problème fondamental, puisqu'il fait partie de la séquence de la méthode. Ce sujet sera repris en extension dans des sections ultérieures mais pour le moment on se limitera à signaler qu'il représente un point faible et que, dans certains cas il peut affecter gravement la stabilité et la convergence de la méthode.

Voici deux graphes associés aux démonstrations 3.1 et 3.2.



graphe 3.1



Graph 3.2

Quoique les auteurs disent, ce qui est vrai d'ailleurs, que le programme de minimisation unidimensionnelle n'est pas central dans la méthode, on vient de démontrer que, indépendamment des arrondis de calcul propre à tout calculateur numérique, la précision inhérent du programme pour situer le minimum sur chaque direction est en rapport avec la probabilité de prendre le chemin vers la stabilité et la convergence quadratique ou le chemin vers le doute et, ce qui est grave, que la possibilité de prendre ce dernier, croît à proximité du minimum.

4.2 - La réalisation de 500 pose un problème pratique. Notre objectif est d'écrire l'algorithme de Davidon comme un sous-programme général qui retourne automatiquement le contrôle au programme principal (programme d'appel) au moment où le meilleur minimum soit situé, sans permettre à la machine de fonctionner plus de temps que nécessaire (par exemple calculant des valeurs oscillantes autour d'un minimum), s'il arrive des situations comme celle décrite au paragraphe 4.1.

L'organisation des programmes sera la suivante :

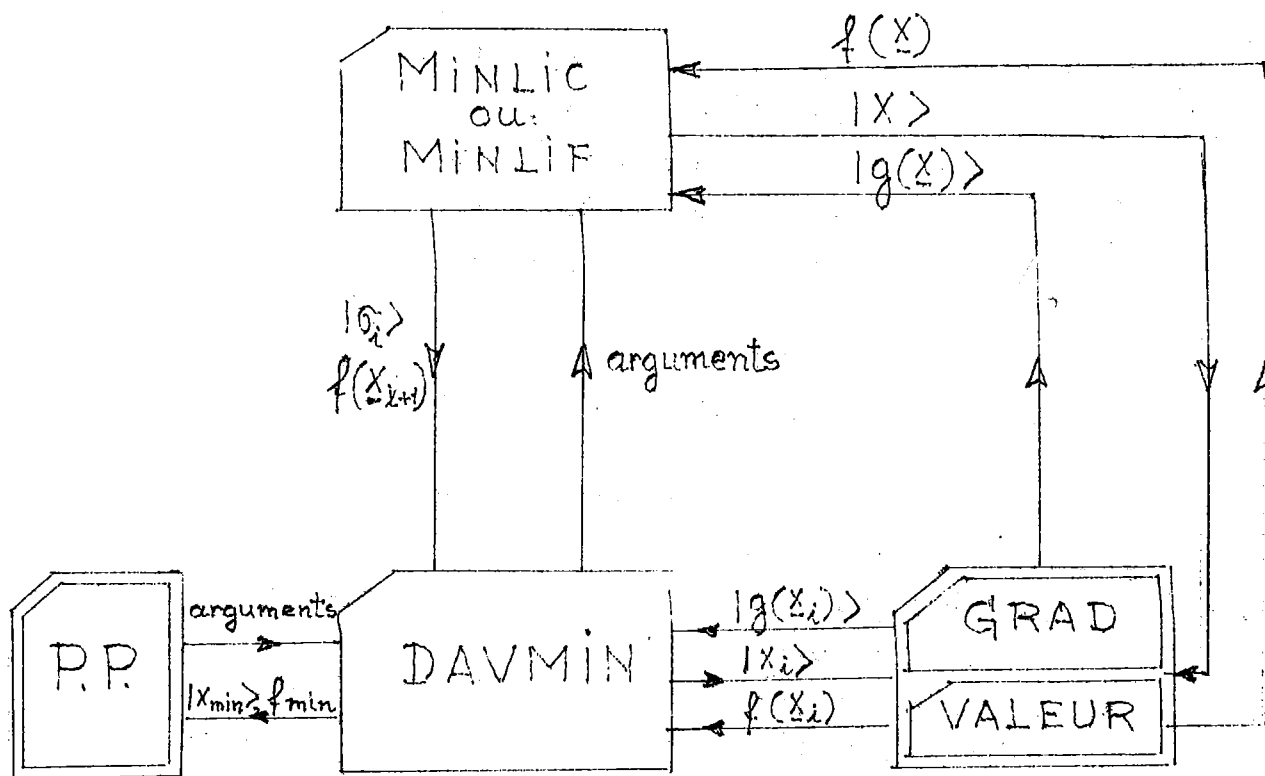


Figure 2

P.P. : programme principal (ou programme d'appel)
 DAVMIN : programme de minimisation Davidon
 MINLIF ou MINLIC : programme M.M.U.
 GRAD : programme de calcul du vecteur gradient
 VALEUR : programme de calcul de valeur de la fonction

P.P., GRAD et VALEUR sont particuliers à chaque utilisation.

III - ANALYSE DE LA MINIMISATION UNIDIMENSIONNELLE DANS LE CONTEXTE DE LA MINIMISATION A N DIMENSIONS

5 - Davidon

La M.M.U. de Davidon est un exemple, optimum d'ailleurs, de la classe de toutes celles d'approximation par une courbe.

On suppose que le profil des valeurs de la fonction sur la direction de recherche est comme celui représenté par la figure 3.

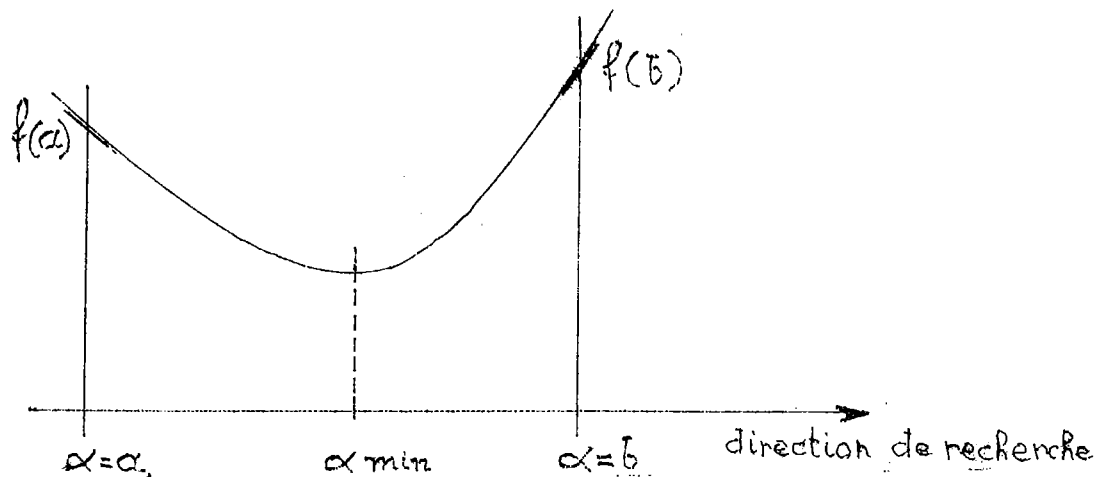


Figure 3

Si la valeur optimum est située à l'intérieur de l'intervalle choisi :

connaissant $f(a)$, $f(b)$ et les pentes de la fonction sur cette direction aux points a et b (gradient projeté sur la direction) Davidon interpole avec la courbe la moins abrupte qui satisfasse ces conditions de frontière. C'est la courbe qui minimise :

$$\int_a^b \left(\frac{d^2 f}{d\alpha^2} \right)^2 d\alpha$$

En fait Davidon fait les hypothèses suivantes :

- 1°) Un intervalle initial
- 2°) La fonction est unimodale tout au long de chaque direction.

La méthode est adaptative dans le sens qu'elle profite de l'information obtenue à chaque mesure pour réduire ou amplifier l'intervalle, s'approchant toujours du point optimum.

Méthode :

- 1 - Pas initial = h
- 2 - Déplacement et amplification de l'intervalle.

On examine f' (dérivée de f) aux points $\alpha = 0, h, 2h, 4h, \dots$ a, b où α est doublée à chaque fois et où b est le premier point pour lequel f' est non négative ou f ne décroît pas

$$\implies \alpha < \alpha_{\min} \leq b$$

3. Interpolation cubique *

On définit :

$$Z = 3 \frac{f(a) - f(b)}{b - a} + f'(a) + f'(b) \quad (33)$$

$$W = (Z^2 - f'(a) f'(b))^{1/2} \quad (34)$$

$$\alpha_e = b - \left(\frac{f'(b) + W - Z}{f'(b) - f'(a) + 2W} \right) (b - a) \quad (35)$$

est une estimation de α_{\min} .

Si $f(\alpha_e)$ est plus petite que $f(a)$ et que $f(b)$ on accepte $\alpha_{\min} = \alpha_e$. Autrement on vient à 4.

* L'interpolation cubique est valable en principe (d'après Davidon, stade 2) pour les cas : $(f'(a) < 0 ; f(b) > f(a) ; f'(b) \text{ quelconque})$ et $(f'(a) < 0 ; f(b) < f(a) ; f'(b) > 0)$. Or Fletcher et Powell définissent le stade 4, lequel fait l'interpolation seulement lorsque : $(f'(a) < 0 ; f(b) \text{ quelconque} ; f'(b) > 0)$ ou bien lorsque : $(f'(a) < 0 ; f(b) \text{ quelconque} ; f'(b) \text{ quelconque})$ ce qui est en contradiction avec le stade 2 de Davidon utilisé aussi par eux (d'après l'algorithme ci-dessus). Pour être logiques, nous allons considérer que l'on effectue l'interpolation seulement dans le cas suivant :

$$(f'(a) < 0 ; f(b) \text{ quelconque} ; f'(b) > 0)$$

et c'est en ce sens que le lecteur doit considérer modifié dorénavant le stade 2.

4 - Réduction de l'intervalle

Si $f'(\alpha_e) > 0$ on revient à 3. pour l'intervalle (a, α_e)

Si $f'(\alpha_e) < 0$ on revient à 3. pour l'intervalle (α_e, b) .

6 - Fibonacci

Fibonacci ou plutôt Kiefer (1953) suppose au départ :

1°) - Un intervalle initial

2°) - La fonction unimodale (non nécessairement continue) tout au long dudit intervalle

3°) - L'optimum est situé à l'intérieur de l'intervalle considéré.

et établit un plan minima de recherche qui consiste à situer séquentiellement les mesures de façon à ce qu'à la fin d'exécution le point minimum reste à l'intérieur d'un intervalle appelé d'incertitude (donc de précision) qui est fixé a priori aussi petit que l'on veut.

La méthode peut jouer avec trois variables :

- intervalle initial
- nombre de mesures
- précision désirée.

En fixant deux quelconques d'entre elles automatiquement est fixée la troisième. Par exemple, avec 24 mesures de la fonction on réduit l'intervalle d'incertitude dans une proportion de l'ordre de 10^{-5} .

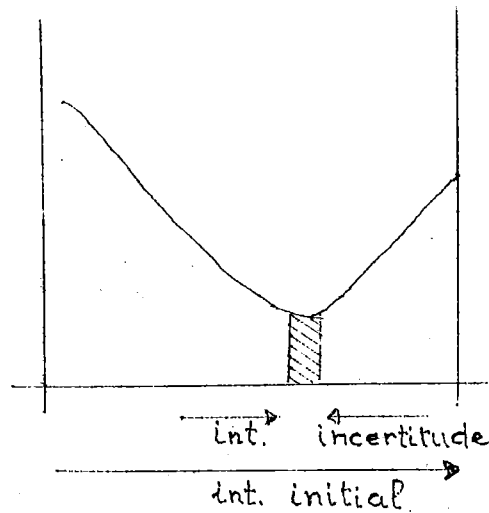


Figure 4.

La méthode de Fibonacci paraît meilleure, a priori, que celle de Davidon et que n'importe quelle autre que l'on puisse imaginer basée sur une interpolation quelconque, et la raison pour ceci, croyons-nous, il faut la chercher dans la sensation de sécurité que produit le fait de pouvoir contrôler à volonté une limite maximum de l'erreur. Cette supposition nécessite pourtant être nuancée soigneusement et éventuellement confirmée.

7 - Analyse en parallèle de Davidon et Fibonacci

Pour dire qu'une méthode est relativement meilleure qu'une autre il faut :

- 1°) que toutes les deux aient au départ les mêmes hypothèses.
- 2°) un critère de comparaison.
- 3°) qu'appliquées aux mêmes cas et dans les mêmes circonstances le critère soit toujours favorable à une par rapport à l'autre.

Il est seulement possible d'effectuer une comparaison absolue lorsque, en plus, tout ce qui vient d'être dit est vrai pour tous les critères possibles.

Nous savons déjà qu'aucune des deux méthodes est meilleure que l'autre, mais malgré tout nous allons faire un essai de comparaison * pour démontrer que toutes les deux, à cause des contraintes inhérentes, peuvent produire des "pannes" situées dans le contexte d'une minimisation à n dimensions.

* Au cours de nos essais nous avons obtenu des résultats parfois très différents en modifiant la politique de détermination des intervalles, en utilisant MINLIF ou MINLIC, en modifiant ou non modifiant ces deux programmes, en supposant connue ou non connue une estimation de la valeur minimum dont on cherche la situation dans l'espace, etc.. Tout ceci pour une même fonction et un même point de départ.

En considérant d'autres fonctions ou d'autres points de départ ou les deux et toutes choses égales par ailleurs, utilisant successivement MINLIC et MINLIF, on obtient des dynamiques de convergence différentes. Et ceci pour une même méthode en changeant les paramètres dont elle dépend.

Or A. Leon dans "A comparison among eight Known optimizing procedures" publié dans le livre "Recent Advances in Optimization Techniques" édité par A. Lavi et T.P. Vogel (J. Wiley 1966) compare huit méthodes, parmi lesquelles Davidon, et tire des conclusions. Il compare toutes ces méthodes, dont les programmes sont en général fournis par divers auteurs, sur cinq fonctions à deux variables et il pondère les résultats à l'aide d'un certain critère. Il va sans dire que, même s'il classifie comme meilleur l'algorithme de Davidon sous le nom de VARMINT cette expérience nous paraît assez discutable.

Préalablement, il faudrait établir une parité des conditions au départ entre Fibonacci et Davidon. Davidon ne fait pas l'hypothèse 3 de Fibonacci mais au contraire amplifie l'intervalle (ou plutôt le déplace) si nécessaire. Fibonacci, lui, réalise imperturbable son plan optimum de mesures et localise le minimum avec la précision voulue. Dans la pratique d'une méthode itérative de recherche à n dimensions, il est aussi probable que le minimum du profil de la fonction au long d'une certaine direction tombe à l'intérieur comme à l'extérieur de l'intervalle initial choisi, à moins que celui-ci soit très grand. Dans ces conditions Fibonacci commence à ne pas être optimum puisqu'il ne localise pas toujours le minimum désiré, à cause de la contrainte n° 3.

7.1 - Nécessité de stratégies supplémentaires

Supposons le profil de la figure 5 et un intervalle initial h_1 pour la recherche Fibonacci. Le minimum localisé est à l'intérieur d'un intervalle d'incertitude collé à l'extrême frontière de l'intervalle initial.

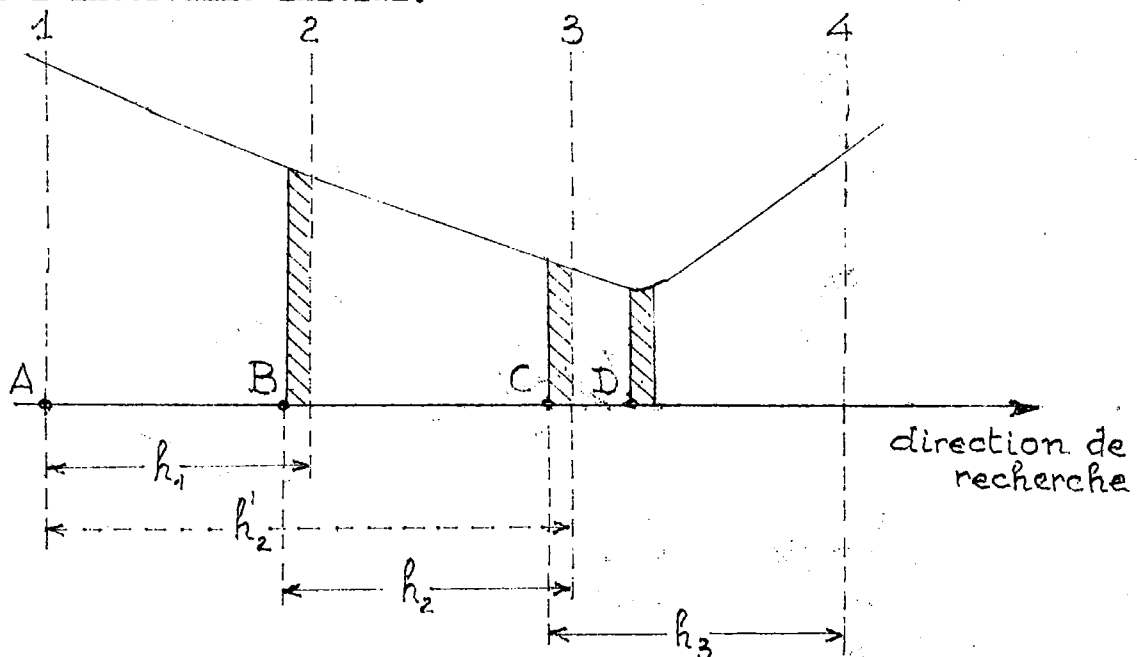


Figure 5

On peut choisir une stratégie qui consisterait à amplifier l'intervalle à h_2 en conservant comme point de départ le point A, mais cela ne serait pas payant. Une autre plus logique, qu'on peut appeler stratégie n° 1, consisterait à partir de B et utiliser un autre intervalle h_2 qui permettra de situer le minimum entre B et l'extrémité de h_2 . Finalement on arrive au minimum D dans la troisième application. Pour fixer les idées supposons que l'on désire une précision de l'ordre de 10^{-5} et choisissons par exemple $h_1 = h_2 = h_3 \dots = 1$, chaque recherche nous a mené à faire 24 mesures, le total de mesures pour localiser D serait donc de 72. Ceci n'est qu'un exemple, bien sûr, dont le résultat est très lié à la longueur de h , à la politique de modification de h et à l'emplacement du minimum D par rapport au point initial A.

7.11 - Une nouvelle stratégie

Nous proposons une stratégie n° 2, qui profite au maximum de l'hypothèse d'unimodalité, lorsque le minimum n'est pas dans l'intervalle. Elle a l'avantage supplémentaire de ne faire a priori aucune supposition sur le côté duquel est situé le minimum.

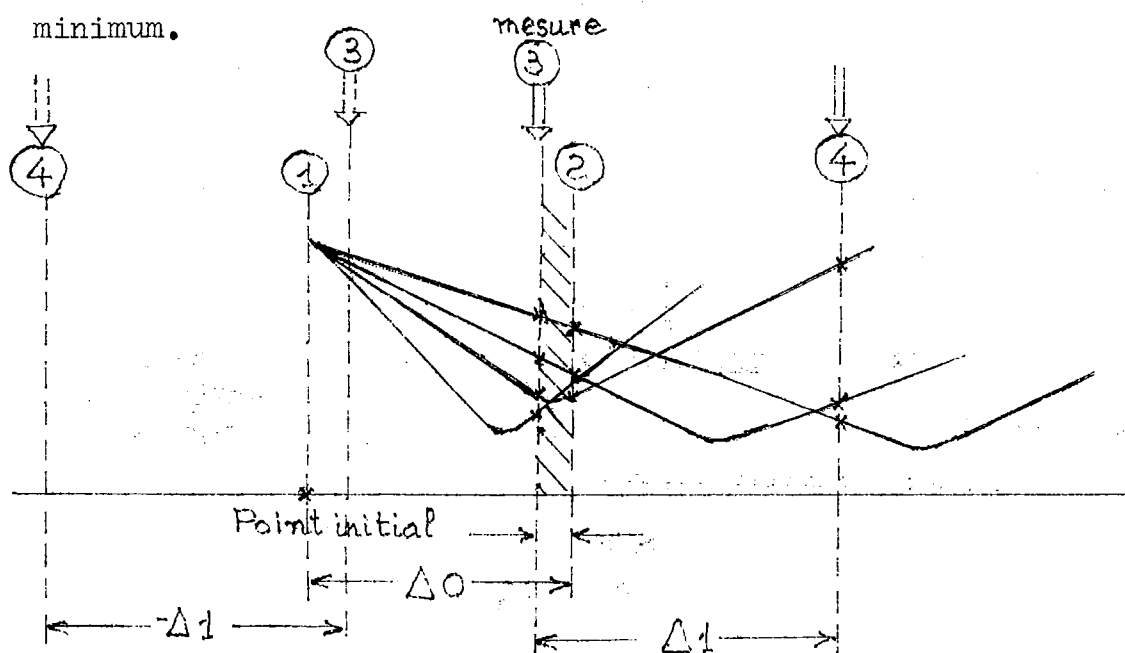


figure 6

Elle consiste à faire les mesures ① , ② et ③ , avec ③ située à l'intérieur de l'intervalle sur un point séparé de ① ou de ② par une distance égale à la précision désirée. De ces mesures résulte la décision de prendre un autre intervalle ou de rester dans le même, d'après l'organigramme suivant :

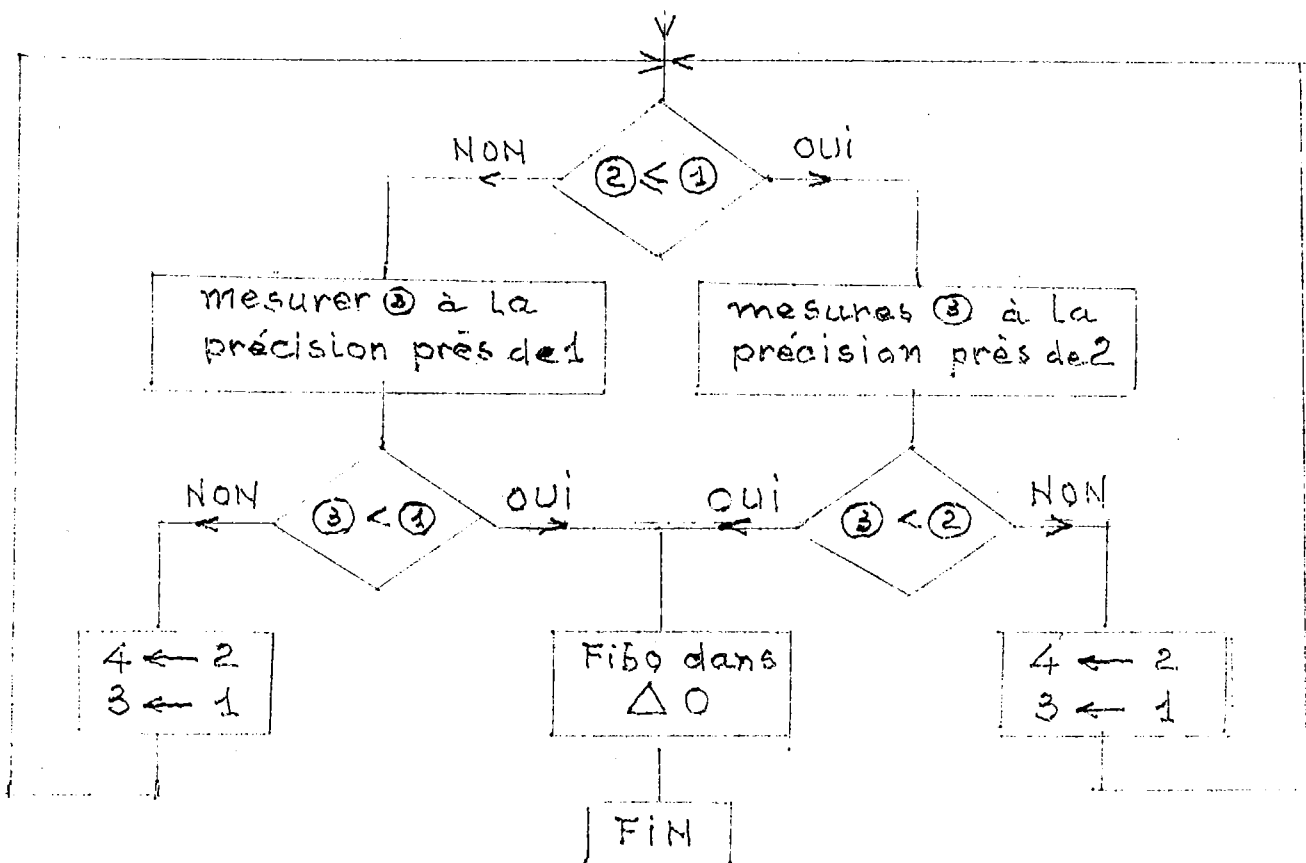


Figure 7

Les chiffres entourés représentent ici des valeurs de fonction. Dans le cas contraire, il s'agit des points sur l'intervalle. La partie symétrique gauche n'est pas nécessaire lorsque l'on sait que le minimum tombe du côté droit du point initial (d'après la figure 6).

Cette stratégie nous aurait situé le minimum, dans le cas de la figure 5, après 30 mesures.

L'avantage d'une stratégie par rapport à l'autre (vis à vis du nombre de mesures) peut être chiffré moyennant certaines hypothèses. Par exemple, imaginons que tous les intervalles sont de la même longueur.

Pour simplifier encore, supposons en plus que l'on veut une précision telle que pour la longueur de l'intervalle considéré il est nécessaire d'effectuer 24 mesures (soit par exemple intervalle unité, précision environ 10^{-5}). Voici un tableau comparatif du nombre de mesures en fonction du nombre de fois qu'on utilise chaque stratégie :

Nombre de fois	Stratégie 1	Stratégie 2
1	24	$24 + 1 \times 2$
2	$24 + 1 \times 24$	$24 + 2 \times 2$
3	$24 + 2 \times 24$	$24 + 3 \times 2$
n	$24 + (N'-1) \times 24$	$24 + N' \times 2$

C'est-à-dire, pour $1 < N' < 2$ la stratégie 2 commence à être plus avantageuse et la différence vaut $22 N' - 24$.

L'expression de la différence entre le nombre de mesures favorable à la stratégie 2 peut être généralisée, si l'on conserve l'hypothèse simple des intervalles toujours de la même longueur.

Nous supposons l'algorithme de Fibonacci utilisée au cours d'une iteration d'un M.M.N. Soient :

N : nombre total d'iterations

N' : nombre moyen de fois, par iteration à N , que l'on utilise la stratégie 1 ou 2.

Θ : nombre de mesures du plan Fibonacci pour l'intervalle et la précision affichés.

Différence du nombre de mesures favorable à la stratégie

$$2 = [(\Theta - 2) N' - \Theta] N$$

La stratégie 2 est favorable pour $N' \geq \frac{\Theta}{\Theta - 2}$

Fixons numériquement les idées, en prenant un cas tout à fait modeste :

pour $N = 15$; $N' = 2$, $\Theta = 24$ la différence est de 300 mesures. En résumé, la stratégie 1 mesure la fonction 720 fois et la stratégie 2 : 420.

Plus concrètement, dans le cas où $\Theta = 24$, la rentabilité de la stratégie 2 est assurée, indépendamment de N , pour $N' \geq 1,08$.

N'importe comment, en utilisant une stratégie ou l'autre, on enlève la contrainte n°3 qui pèse sur la recherche Fibonacci et à ce moment là Davidon et Fibonacci sont soumis aux mêmes contraintes vis à vis de ce type d'applications, à savoir :

- 1) un intervalle initial
- 2) la fonction doit être unimodale au long de cet intervalle.

Maintenant on peut faire un essai de comparaison :

- Au niveau de recherche unidimensionnelle dans un espace à une dimension on peut considérer comme supérieur Fibonacci dû à sa précision.
- Si l'on établit comme critère la longueur du programme, Davidon est un peu plus court.
- Au niveau de recherche unidimensionnelle dans le contexte d'une recherche dans un espace à n dimensions on ne peut rien dire. Et en plus, si l'on prend comme critère le nombre de mesures de la fonction, la comparaison est trompeuse, car il ne faut pas oublier que Davidon nécessite le calcul de la valeur de la fonction, de son vecteur **gradient** ainsi que le produit scalaire du gradient et de la direction. Or, si l'on admet que le temps de calcul de chaque composant du gradient est au moins de l'ordre de celui de calcul de la fonction et très souvent supérieur, il faut reconnaître que la mesure à chaque point équivaut, dans Davidon, à $(n + 1) F$ (si l'on appelle F le temps de calcul de la valeur de la fonction en un point). D'où l'on déduit un inconvénient caché inhérent à la M.M.U. de Davidon :

Le temps de calcul nécessaire à la méthode croît presque linéairement avec le nombre de dimensions (le nombre des variables indépendantes) de l'espace dans lequel est définie la direction de recherche.

Voici un exemple illustratif :

Soit 5 le nombre de points qui, dans des circonstances déterminées, nécessite la M.M.U. de Davidon pour situer son estimation d'un minimum dans un espace :

1°) à 5 dimensions	\implies	5 (5 + 1) F = 30 F
2°) à 10 dimensions	\implies	5 (10 + 1) F = 55 F
3°) à 15 dimensions	\implies	5 (15 + 1) F = 80 F

La M.M.U. Fibonacci, au contraire, est invariante quant au nombre de mesures de la fonction vis à vis du nombre des dimensions de l'espace. Il est facile à imaginer ce qu'on pourrait faire avec Fibonacci + Stratégie 2 disposant de 80 mesures. En vérité il faut considérer aussi que, très souvent, un algorithme de programmation non-linéaire à n dimensions à besoin du calcul du gradient au point initial pour déterminer la direction de recherche, donc par ce fait il y a lieu d'ajouter à Fibonacci + S.2 un temps N'F.

7.2 - Conclusions provisoires.

(valables pour des fonctions absolument unimodales)

- a) Il est très probable que Fibo + S.2 soit supérieur à Davidon dans un espace d'un nombre élevé de dimensions (par exemple à partir de 8). Critère : le temps de calcul.

- b) Si le critère est la sécurité, Fibonacci est supérieur.
- c) Dans le cas de problèmes à un grand nombre de variables, il serait probablement intéressant d'utiliser au lieu de la M.M.U. de Davidon, un autre type d'interpolation qui ne nécessite pas le calcul des dérivées. Critère : le temps de calcul.
- d) Si la dimension de l'espace n'est pas très élevée, on ne peut rien conclure a priori. (Il faut rappeler que le contexte est une approximation au second degré ou hyperquadratique. On ne doit pas oublier non plus que ce qui compte est le nombre total de fois que l'on calcule la fonction au cours de la résolution du problème). En fait, l'expérience nous montre que parfois Davidon est meilleur que Fibonacci ou le contraire, selon la fonction, selon le point initial.

Les seules conclusions positives sont a) et b), le reste ne peut nous servir que de guide relatif quant au choix de l'une ou l'autre M.M.U. ; on verra ceci plus clairement à la lumière de nouvelles considérations sur les contraintes 1 et 2, qui conditionnent l'efficacité absolue de nos algorithmes de P.N.L. et qui, nous verrons, imposeront une révision de quelques unes des conclusions que l'on vient de dresser, en particulier celles qui paraissent les plus positives.

8 - Fonctions unimodales, mais ^{pas} fortement unimodales.

Nous supposons que, dans le domaine de l'espace où se développe notre recherche, la fonction considérée est unimodale, c'est-à-dire a seulement un point extrême.

Définition 1 (Wilde, [4] , page 93)

Une fonction $f(\underline{X})$ est unimodale si, quel que soit le point $|a\rangle$ de la région expérimentale, il existe un chemin strictement croissant joignant $|a\rangle$ à l'extremum $|X\rangle$.

La fonction représentée à la figure 8 est unimodale.

Définition 2 ([4] , page 94)

Une fonction est dite fortement unimodale si la droite joignant au sommet un point $|a\rangle$ quelconque de la région expérimentale est un chemin croissant.

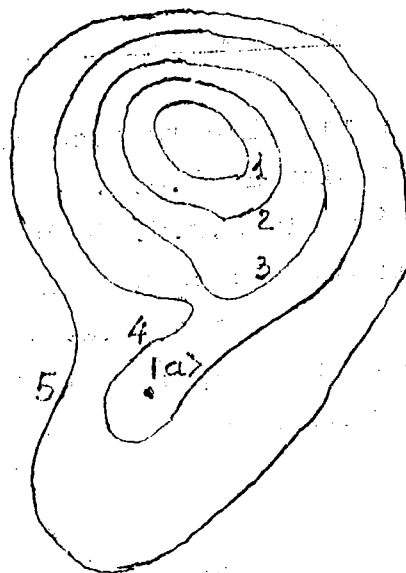


Figure 8

La fonction de la figure 8 n'est pas fortement unimodale.
La différence entre une définition et autre tient aux mots chemin et droite.

Or, dans nos programmes nous utilisons des droites et la politique consiste à ⁿes diriger vers le sommet. La situation qui se pose est la suivante :

- Les M.M.U. ne sont complètement valables que dans des zones fortement unimodales.
- Par conséquent si l'on veut pouvoir les utiliser pour des fonctions générales, il faudra y introduire des modifications. La seule politique possible consiste à essayer d'éliminer dans M.M.U. la contrainte d'unimodalité.

9 - Discussion de la contrainte d'unimodalité dans M.M.U.

A la section 4 nous avons attiré l'attention sur l'importance capitale du rôle de la précision de M.M.U. dans la théorie de la convergence et stabilité de la M.M.U. de Davidon. Dans cette section et les suivantes on reprend cette critique et on la relationne avec les deux contraintes de M.M.U. et avec la réalisation d'un test de convergence, dans le but d'augmenter la fiabilité de la M.M.N. de Davidon et de toutes celles basées sur les M.M.U.

Pour fixer les idées on va prendre comme base graphique de raisonnement la fonction du 2^e degré, dite de Rosenbrock, test habituel de tous les programmes de minimisation, en vertu de sa vallée profonde et longue, qui suit la courbe $X_1^2 = X_2$.

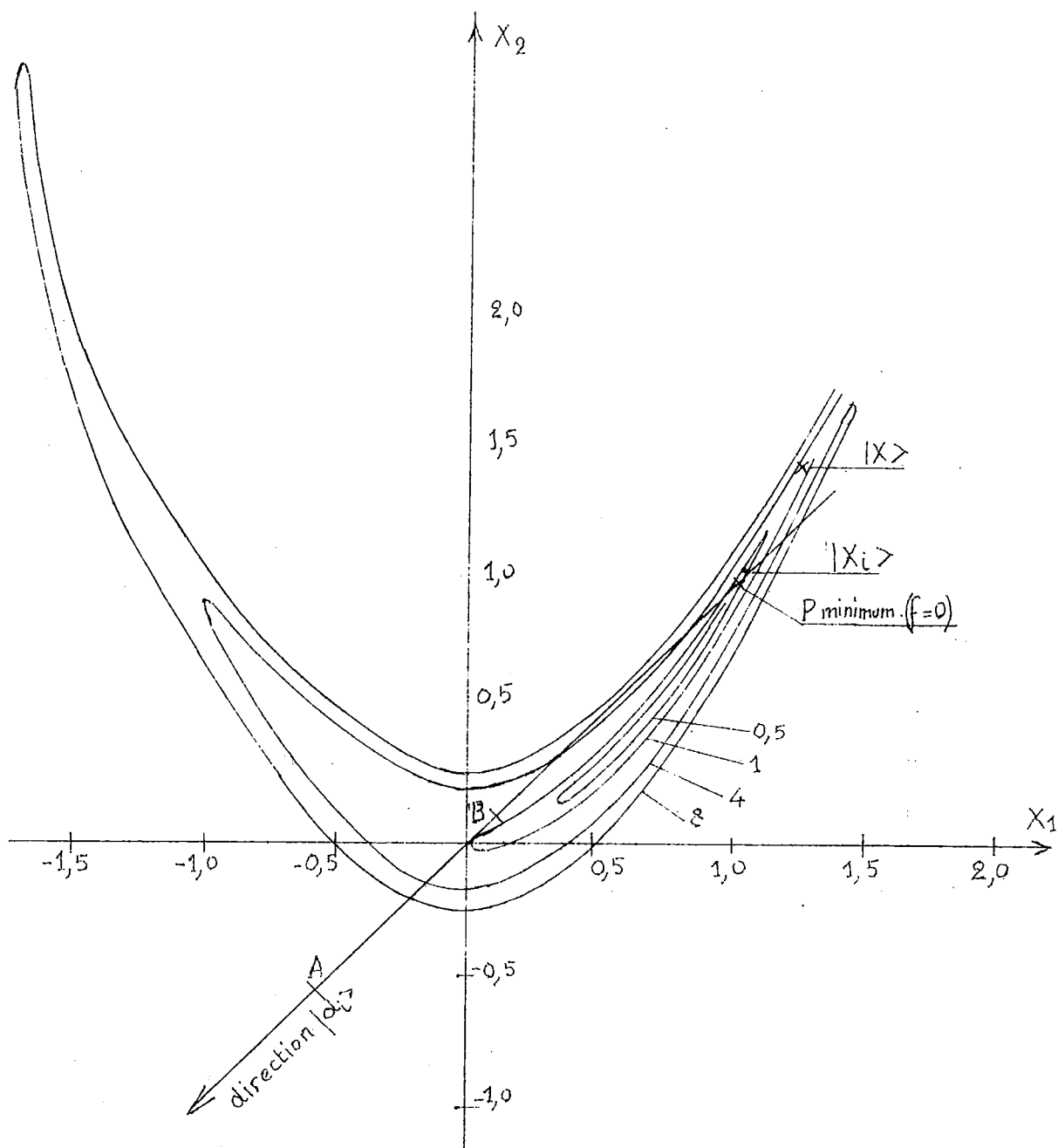


FIGURE 9

L'expression de cette fonction est :

$$f(X_1, X_2) = 100 (X_2 - X_1^2)^2 + (1 - X_1)^2$$

et ses contours de niveau sont ceux de la figure 9. Le minimum est situé en (1,1) et vaut zéro.

Pour le moment nous allons supposer que ce qui se produit aux points éloignés du minimum ne nous intéresse pas. Imaginons que le point $|X_i\rangle$, localisé au cours de l'iteration $i - 1$, est le point de départ d'une recherche unidimensionnelle le long de la direction $|d_i\rangle$, déterminée par l'algorithme pour l'iteration i . Nous avons supposé que le point $|X_i\rangle$ est situé très près du point optimum et que la fonction pour ce point prend une valeur très proche de zéro (par exemple 10^{-8}).

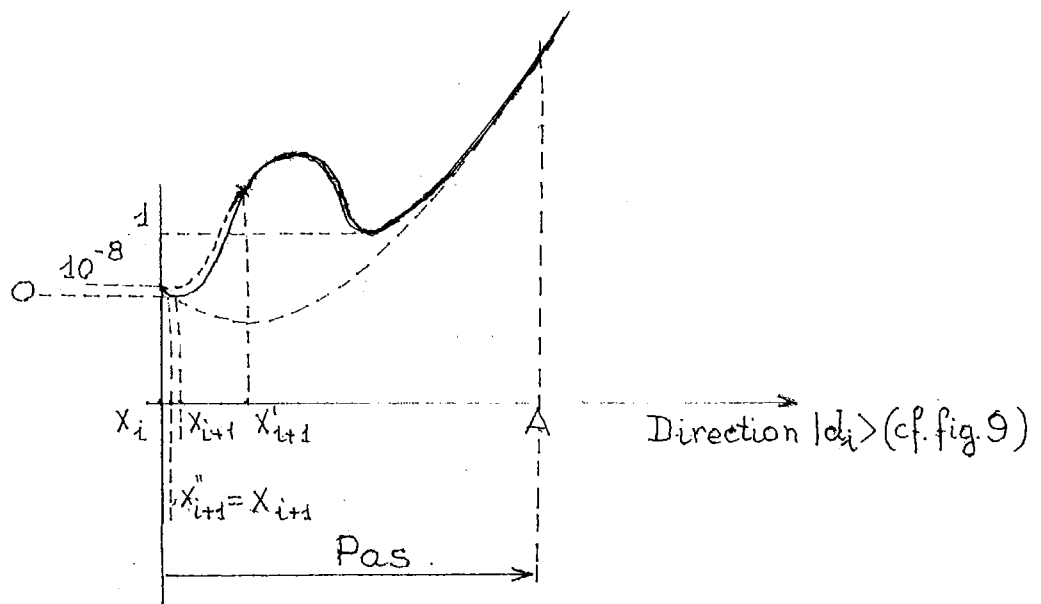
La fonction n'est pas unimodale sur la direction choisie qui, dans le cas concret de la figure 9, est celle qui passe par le minimum.

Les exemples suivants, entre plusieurs possibles, ont pour but de mettre en relief graphiquement les termes dans lesquels se pose le problème défini à la section 8.

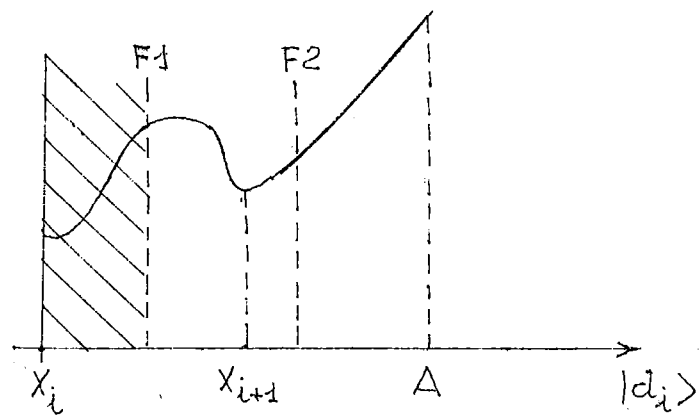
1°) L'intervalle initial est $\| |A\rangle - |X_i\rangle \|^{1/2}$

Davidon (figure 10 a)

On a dessiné un cas relativement favorable pour lequel Davidon converge sur un point $|X_{i+1}\rangle = |X''_{i+1}\rangle$ mieux situé que $|X_i\rangle$ après deux interpolations cubiques. Très différent aurait été le résultat si $|X'_{i+1}\rangle$ (point estimé) serait tombé à droite du maximum local dans le profil de la figure 10 a.



a) Davidson



b) Fibonacci

Figures 10 a et b

Fibonacci (figure 10 b.

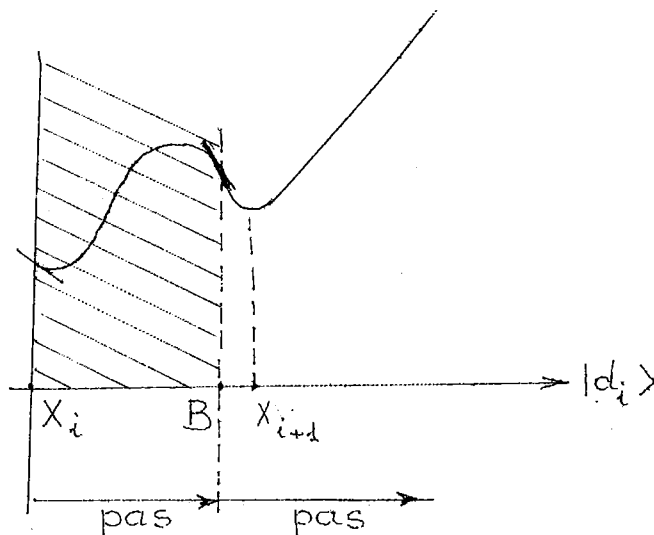
(Sur l'organisation du plan Fibonacci cf Wilde [4] ou Recio [5])

Si les deux mesures du plan sont F1 et F2, le résultat est l'élimination de la région hachurée, donc du vrai minimum ;

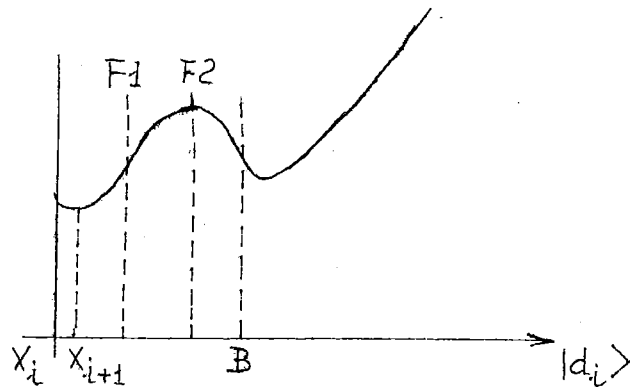
$|x_{i+1}\rangle$ correspond à un point qui, dans le cas de la figure 9, est très éloigné du vrai minimum.

C'est une conséquence malheureuse inhérente au plan Fibonacci. Le cas des figures 9 et 10b n'est pas seulement une supposition mais un fait réel qui nous est arrivé pendant la mise au point de nos programmes. Dans un tel cas l'algorithme peut produire une séquence oscillatoire des valeurs de la fonction autour du minimum.

2°) L'intervalle initial est $\| |B\rangle - |x_i\rangle \|^{1/2}$



a) Davidon



b) Fibonacci

Figures 11a et b

Davidon

Maintenant c'est Davidon qui situe $|x_{i+1}>$ loin du vrai.

Fibonacci

Par contre Fibonacci situe un minimum correct, dans la mesure de la précision désirée.

Une première façon d'éviter cet inconvénient serait de ne pas accepter $|x_{i+1}>$ que si $f(x_{i+1}) \leq f(x_i)$; autrement on arrête l'algorithme, acceptant $|x_i>$ et $f(x_i)$ comme valeurs finales. Celui-ci est un critère que suivent divers auteurs d'algorithmes à convergence théoriquement continue. Mais si la situation décrite pour $|x_i>$ se présente pour $|x>$ (voir figure 9) et que

nous acceptons le critère précédent, nous avons accepté au même temps comme minimum un point très éloigné du vrai.

Une meilleure tactique à proximité du minimum serait de prendre un intervalle initial petit de façon à augmenter la probabilité d'unimodalité dans la zone de déplacement. Cette manière de raisonner nous amène à considérer le problème de la politique générale du choix des intervalles dans la région de l'expérimentation.

Pour augmenter l'efficacité et la sécurité de l'algorithme, une condition nécessaire est de choisir une politique d'adaptation de l'intervalle de recherche pour M.M.U. telle que celui-ci diminue au fur et à mesure qu'on s'approche du minimum. (Le problème est de choisir cette politique ; voir section 10).

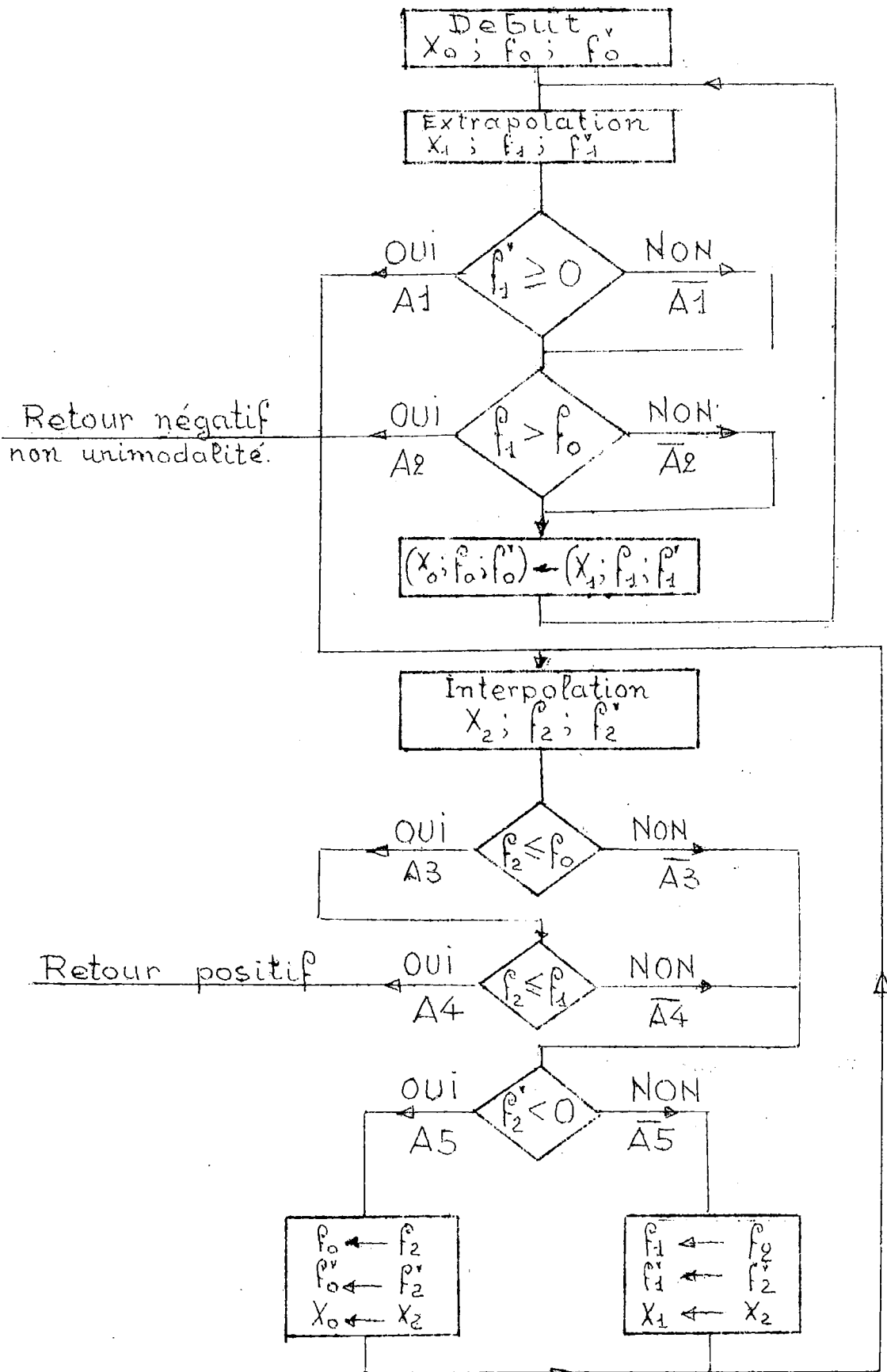
En effet, loin du point minimum ce n'est pas fondamentalement une grande précision (pourvu que $f(\underline{X}_{i+1}) \leq f(\underline{X}_i)$) et l'on peut se contenter de peu de mesures. Le contraire serait antiéconomique.

La deuxième politique possible serait de choisir un intervalle constant et petit (mais quelle longueur ?) et d'utiliser Davidon ou Fibonacci + S.2, mais ceci obligerait à un grand nombre de mesures et forcément à une convergence lente.

La troisième consisterait à choisir un intervalle constant suffisamment grand, ce qui exige un nombre énorme de mesures, puisqu'il faut fixer une fois pour toutes la précision absolue de résolution du problème. Ceci par surcroît, d'après ce qu'on vient de démontrer et contre ce que l'on pouvait attendre, engendre l'insécurité.

ORGANIGRAMME N° 1.

Lignes générales de la M.M.U. de DAVIDON.



Ces deux dernières politiques comportent d'ailleurs le problème tactique du choix optimum de la longueur constante de l'intervalle.

En général, la troisième politique est la pire de toutes.

On va donc supposer qu'on applique une stratégie adaptative du 1er type. Notre but est de nous assurer de l'unimodalité dans la zone immédiate du point qui nous donne la valeur la plus petite de la fonction sur la direction de recherche.

9.1. - Stratégie pour la M.M.U. de Davidon

A toutes les considérations que nous avons faites ajoutons celles-ci :

Voir l'organigramme n°1 qui décrit les grandes lignes de la M.M.U. de Davidon telles que nous les avons interprétées. Il y a là ce que nous appelons les :

séquences qui amènent au retour positif :

$$\text{Retour positif} = (\bar{A}_1, \bar{A}_2)^* \cdot A_1 ((\bar{A}_3 + A_3, \bar{A}_4)(A_5 + \bar{A}_5))^* A_3 \cdot A_4$$

Ceci décrit dans le langage des expressions régulières les séquences qui conduisent au retour positif, c'est-à-dire unimodalité assurée (d'après Davidon, Fletcher et Powell).

Ce que nous appelons retour négatif est la non unimodalité qui impose une prise de précautions.

(signification : chaque lettre signifie la réalisation de la condition exprimée dans le graphe. Ainsi, par exemple, c'est $A1$ si $f_1' \geq 0$. Le symbole $*$ signifie zéro, une, deux ... ∞ itérations des séquences contenues. La concaténation des séquences n'est pas commutative).

Séquences qui pourtant cachent une possible non unimodalité.

La partie hachurée est celle que Davidon élimine. La courbe de traits interrompus est la courbe interpolée.

$(\overline{A1.A2})^* \overline{A1.A2}$ déjà considérée à l'organigramme n° 1

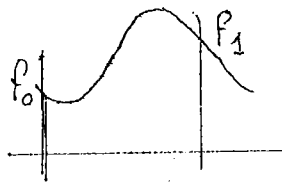


Figure 12

$\overline{A3.A5} ?$

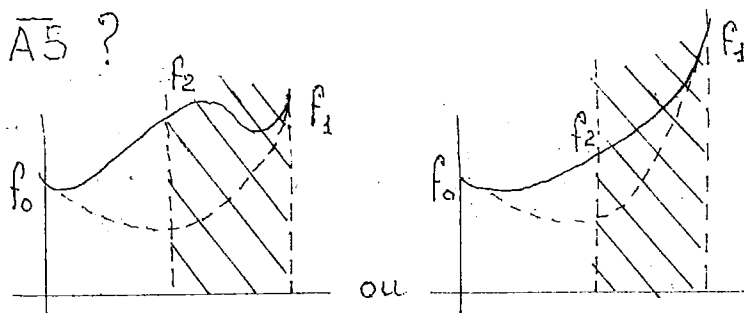


Figure 13

$A3 \cdot \overline{A4} \cdot A5 ?$

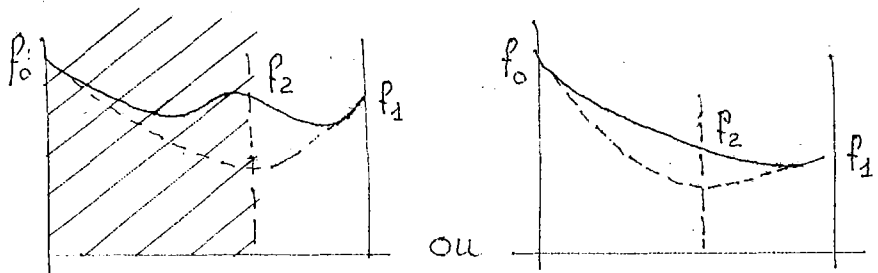


Figure 14

$A3 \cdot \overline{A4} \cdot \overline{A5}$

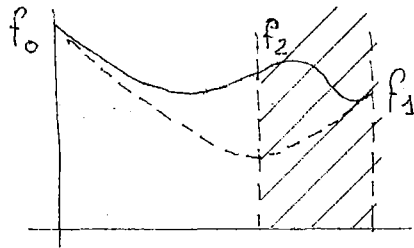


Figure 15

On a déjà vu dans les considérations générales, concrétisées en la surface de Rosenbrock, comment on peut trouver dans certaines directions des profils non unimodaux de la fonction. On comprend aussi que dans quelques cas ($f_2 \leq f_0$; $f_2 \leq f_1$) la méthode peut nous amener à accepter un faux minimum dans un profil non unimodal (figure 16), ce qui pourrait faire générer une matrice S non définie strictement positive, bref à choisir une direction non descente, ou peut-être à osciller autour du minimum.

Maintenant nous voyons en plus, à l'aide de l'analyse des séquences que l'on vient de faire, qu'au moins la bimodalité, peut être détectée dans beaucoup de cas connaissant à chaque instant la séquence que le programme de M.M.U. est en train d'exécuter.

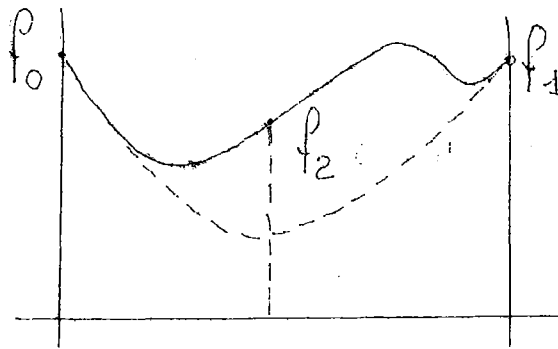


Figure 16

Par exemple de l'étude des figures 12, 13, 14 et 15 l'on déduit que les séquences $\overline{A3.A5}$ et $A3.A4.A5$ indiquent toujours la présence d'une bimodalité.

Au cours de la mise au point de M.M.U. de Davidon, nous avons trouvé les cas suivants :

Fonction : Rosenbrock déjà citée.

1° - point initial $|X_0\rangle = (-1.2, 1)$

Itération 20 ; $|X_1\rangle = (0.9999980927 ; 0.9999967813)$

vecteur direction $|d_1\rangle = (0.15716 \text{ E-05} ; 0.25611 \text{ E-05})$

on obtient la séquence $\overline{A3.A5}$

2° - point initial $|X_0\rangle = (3;3)$

Itération 114 ; $|X_1\rangle = (0.9999985099 ; 0.9999969006)$

vecteur direction $|d_1\rangle = (0.14802 \text{ E-05} ; 0.30778 \text{ E-05})$

on obtient la séquence $A3.\overline{A4.A5}$

Ce qui est est ennuyeux dans les deux cas, c'est que la séquence se boucle (cycle) indéfiniment et le sous-programme ne retourne pas le contrôle au programme d'appel ; ceci est dû à la forme d'interpolation utilisée.

La façon d'éviter cet inconvénient (si nous voulons être logiques avec notre politique) est de produire une réduction de l'intervalle de recherche, prenant comme base le point initial de la recherche. Pour cela, il est nécessaire d'introduire un indicateur pour chaque séquence dangereuse.

Il suffit de deux compteurs :

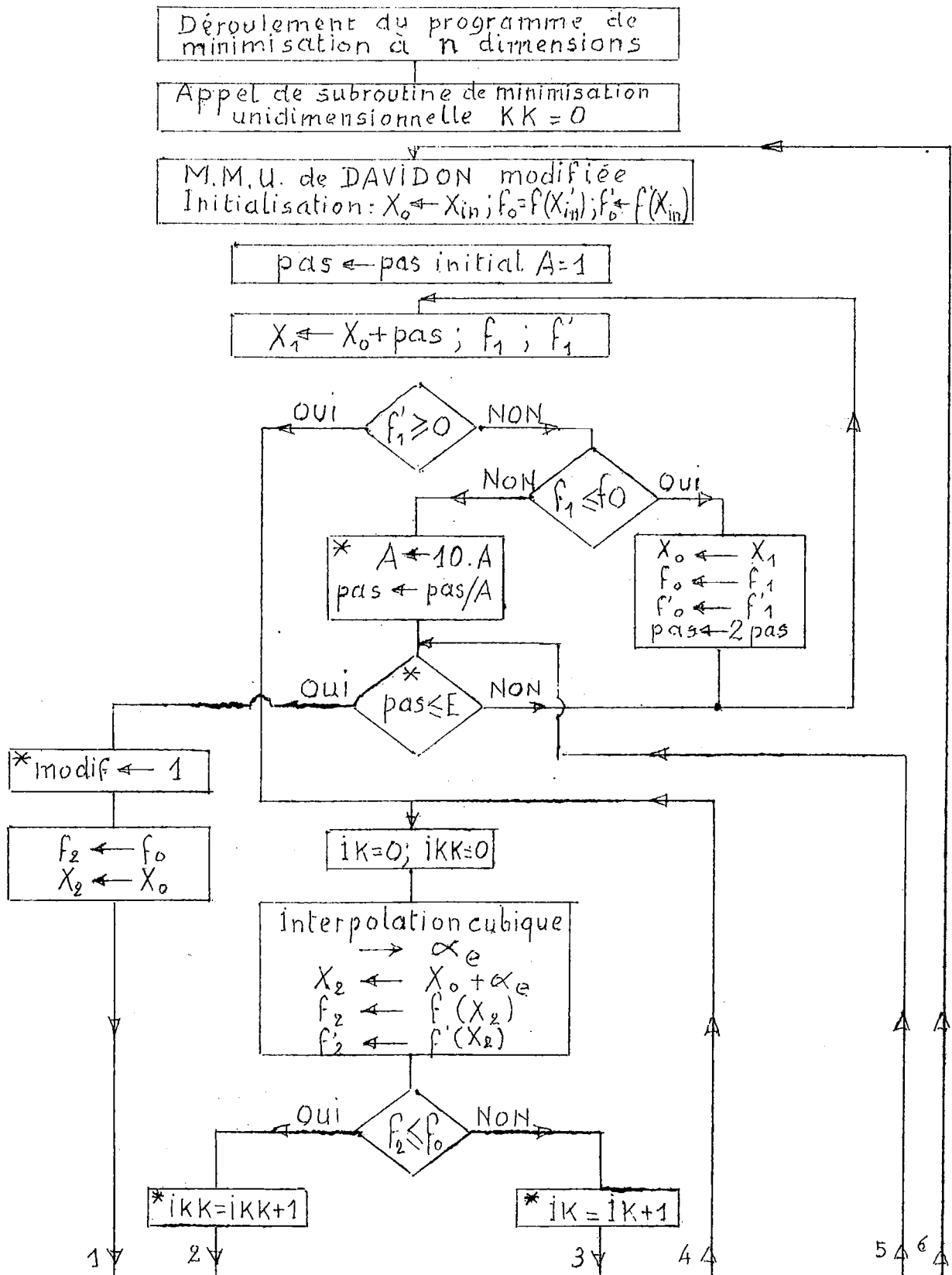
$$\begin{aligned} & \overline{A3} \cdot (IK = 1) \cdot A5 \cdot (IK = 2) \\ & A3 \cdot (IKK = 1) \cdot \overline{A4} \cdot (IKK = 2) \cdot \overline{A5} \cdot (IKK = 3) \end{aligned}$$

L'organigramme n° 2 représente sur M.M.Ü. de Davidon ces modifications et d'autres, toutes repérées par un astérisix. Celles-ci tendent :

- à éviter les pièges de non unimodalité en maintenant la politique de rester près du point initial de la recherche tant qu'on ne trouve pas un point où la valeur de la fonction soit plus petite.
- à s'approcher très vite du point initial avec la précision absolue désirée E , si par un hasard la direction ne serait pas orientée dans le sens descendant.
- à vérifier que le meilleur point obtenu dans la recherche nous assure une matrice S définie positive pour l'itération suivante ($f'_0 < 0$). Au cas contraire réinitialiser le système.

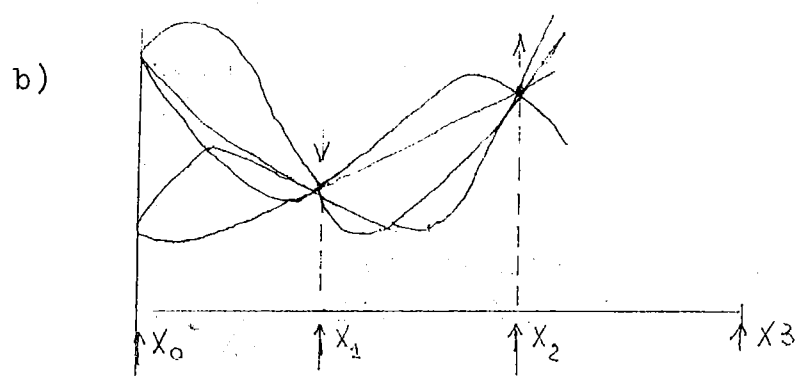
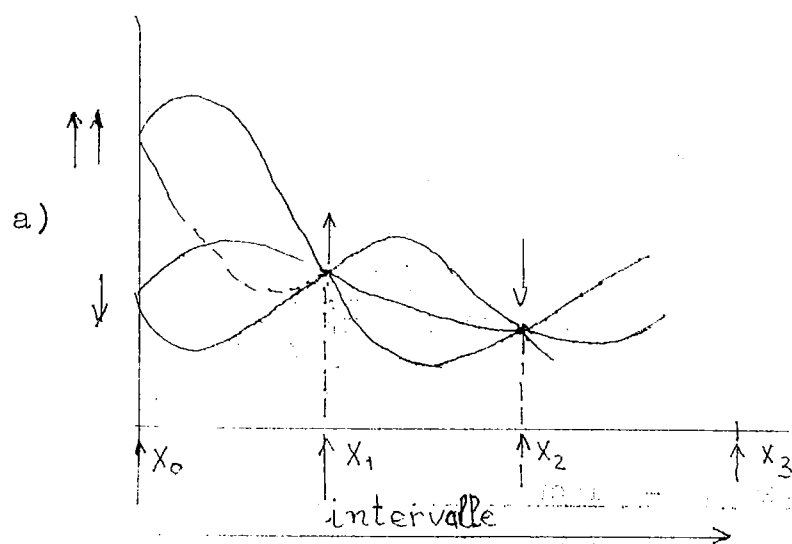
ORGANIGRAMME N°2 (Planche I)

M.M.U. DAVIDON MODIFIÉE



9.2 - Stratégies pour la M.M.U. de Fibonacci.

Plusieurs solutions sont possibles, toutes comportant un certain risque.



Figures 17a et b

Soient X_0, X_1, X_2, X_3 : X_0 et X_3 sont les extrémités de l'intervalle ; X_1 et X_2 les points de mesure choisis par le plan Fibonacci. Notre politique étant conservatrice à proximité de X_0 , dans le but de ne pas perdre ce qu'on a déjà gagné, on ne s'occupera pas des valeurs de la fonction en X_3 .

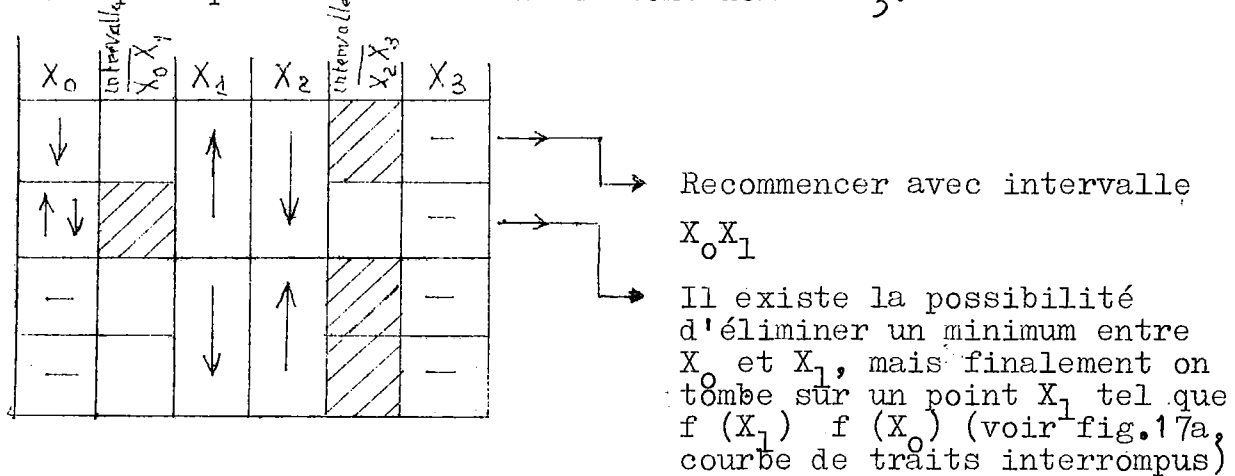


Figure 18 - Fibonacci modifié

Une première modification de l'algorithme de Fibonacci en vue de réduire l'intervalle est expliquée par le tableau de la fig. 18 et les figures 17 a et b. La signification est la suivante :

On prend toujours comme base de raisonnement les mesures aux points X_1 et X_2 ; nous représentons par ↑ la valeur la plus élevée et par ↓ l'autre. La mesure en X_0 se pondère par comparaison avec celle repérée par ↑. Si elle est plus grande (↑↑) si plus petite (↓↓) et si la chose est indifférente par un trait (—). La zone hachurée est la part éliminée de l'intervalle.

Pour amplifier l'intervalle on peut prendre la stratégie n° 1 de la section 7.1, avec une certaine marge de sécurité pour tenir compte des arrondis de la machine.

Une deuxième possibilité consiste à supposer que notre décision traverse toujours des zones unimodales et applique Fibo normal avec S.l.

Une troisième possibilité : appliquer la stratégie n° 2 (S.2) en supposant que la géographie est fortement unimodale. Nous savons que ceci n'est pas vrai mais si le mécanisme de génération d'intervalles d'exploration est raisonnable, ce sera aussi raisonnable de supposer que dans la plupart des itérations la fonction est unimodale au long de l'intervalle initial.

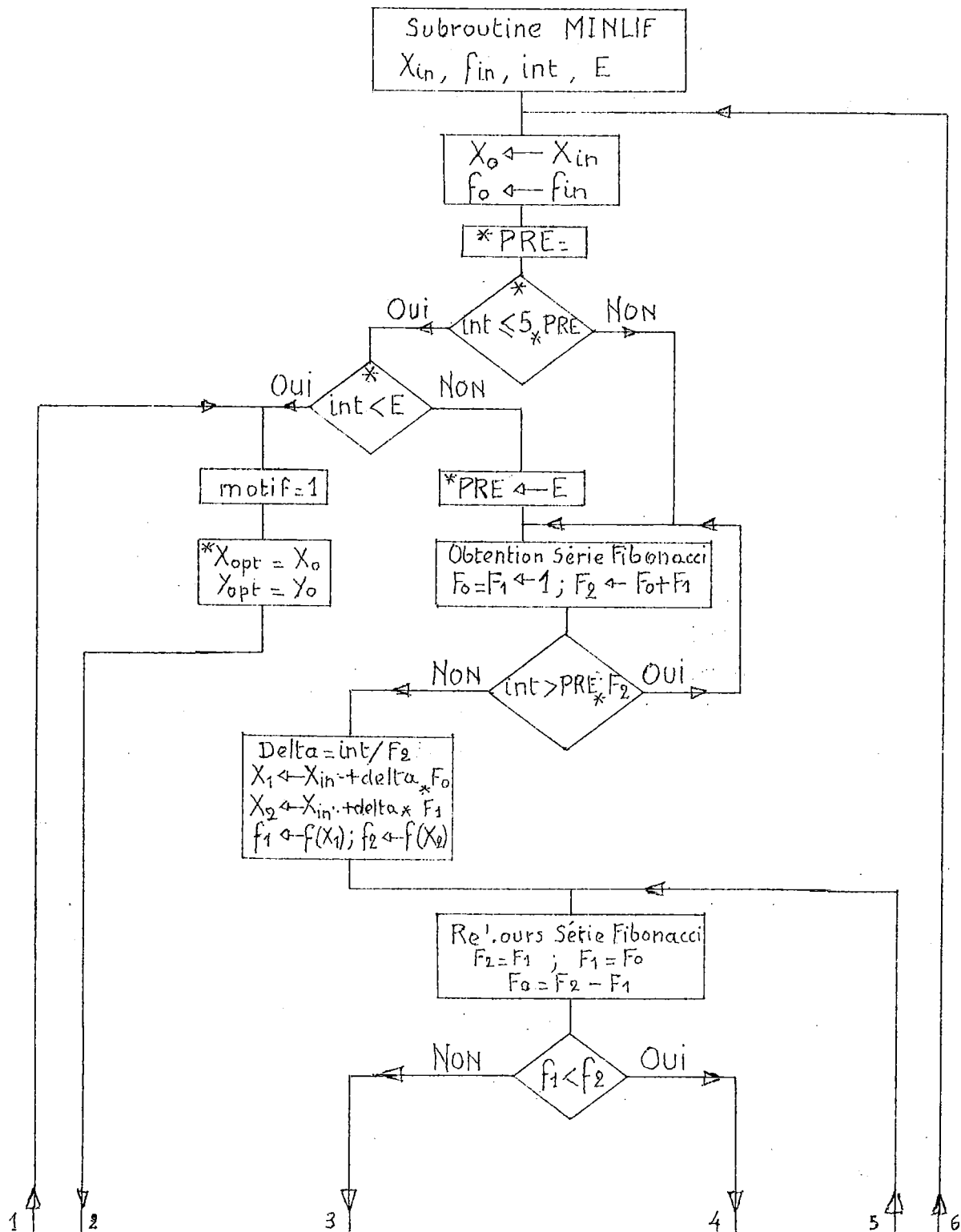
Dans toutes les trois solutions il serait nécessaire d'effectuer un test à la fin de l'exploration : si $f(\underline{X}_{i+1}) \geq f(\underline{X}_i)$, on recommence la recherche sur la même direction, à partir du même point, avec un intervalle initial plus petit que celui de l'exploration précédente.

Trait commun aux trois est l'importance de la valeur en $\{X_i\}$ comme base de référence.

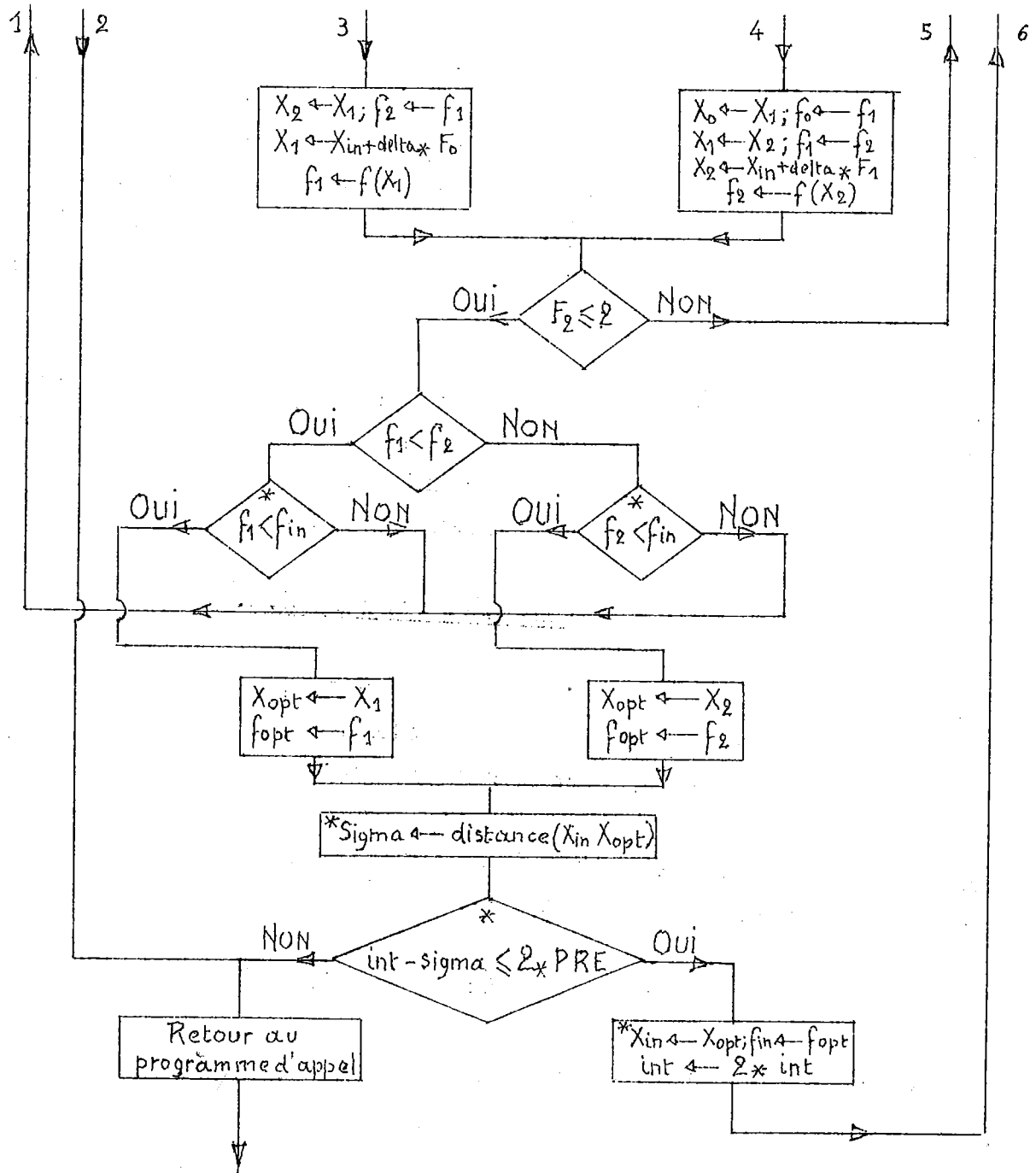
Seulement la première représente une modification de l'algorithme original de Fibonacci ; c'est-à-dire sur la façon de réduire l'intervalle d'incertitude. Les solutions 2 et 3 sont Fibonacci + Stratégie d'expansion respectivement a posteriori et a priori. Il est difficile de choisir parmi les trois, étant donné que dans toutes on prévoit des tests suffisants pour ne pas accepter des oscillations. La possibilité du choix s'établirait finalement entre la 1ère et la 3ème (la 1ère étant nettement plus sûre que la 2ème au prix d'une très légère modification). La 3ème comporte plus de risque mais est peut être plus économique au point de vue nombre total de mesures, surtout dans les cas où la géographie des contours n'est pas excessivement tordue.

Dans le plan Fibonacci (cf Wilde [4] P. 31 ou Recio [5]) le nombre de mesures est lié au rapport: intervalle d'incertitude initial sur l'intervalle d'incertitude (précision PRE) par

ORGANIGRAMME N° 3 (Planche1) M.M.U. FIBONACCI modifiée



ORGANIGRAMME N°3 (Planche 2) M.M.U. FIBONACCI modifiée



l'intermédiaire des termes de la série de Fibonacci de façon à ce que, fixant la précision, le nombre de mesures augmente ou diminue avec la longueur de l'intervalle. Ceci nous permet de commencer la méthode générale en imposant une précision petite pendant les itérations qui se développent loin du minimum, dans le but d'épargner des mesures, et commuter à la précision désirée E lorsque l'intervalle est plus petit d'une certaine valeur, par exemple $5 \times \text{PRE}$ (PRE peut être de deux à trois ordres inférieur à E).

L'organigramme n° 3 présente une solution qui consiste simplement en Fibonacci normal + déplacement de l'intervalle + commutation de précision + retour pour réinitialisation du système lorsque le programme n'a pas réussi à améliorer le point.

10 - Choix de l'intervalle initial dans la M.M.U.

Dans les sections précédentes on a démontré, et les expériences nous l'ont confirmé, que la seule politique vraiment efficace de génération d'intervalles doit être adaptative.

L'on prétend déterminer λ raisonnablement pour situer l'extremum de notre intervalle initial

$$|X_{\text{ext}}\rangle = |X_i\rangle + \lambda |d_i\rangle \quad (36)$$

Deux solutions sont possibles :

10.1 - On connaît la valeur exacte ou estimée du minimum de la fonction.

Ce cas se présente très souvent dans des problèmes d'optimisation et d'identification. Imaginons par exemple, le problème simple d'ajuster par les minimum carrés une courbe à une série de points expérimentaux.

- $S|g_i\rangle = |d_i\rangle$ serait le pas à prendre si $S = A^{-1}$. Ceci n'est pas vrai, en général puisque, à un point quelconque, S n'est qu'une estimation insuffisante de A^{-1} , mais donne une échelle naturelle très valable pour la comparaison ($\lambda = 1$). Mais nous croyons plus général de fixer a priori comme échelle un pas (pas, en vraie grandeur) en rapport logique avec les dimensions de l'espace à explorer. Alors $\lambda = \text{pas}/\text{long}(|d_i\rangle)^{(1)}$; donc $\lambda = 1$ ne serait qu'un cas particulier où $\text{pas} = \text{long}(|d_i\rangle)$.

Très souvent cet intervalle peut être excessif. Cherchons-en un autre pour le calibrer.

Appelons :

f_i : la valeur de la fonction au meilleur point trouvé au cours de l'itération i .

f_{est} : valeur minimum estimée (donnée a priori)

$(f_i - f_{\text{est}})$ représente la quantité par laquelle $f(\underline{X})$ dépasse sa valeur minimum (cf section 2.1) et lorsque $\mathbf{S} \rightarrow A^{-1}$ on peut écrire, d'après équation (12)

(1) $\text{long}(|d_i\rangle)$ signifiant la longueur du vecteur.

$$\frac{1}{2} \langle g_i | S | g_i \rangle = f_i - f_0 \quad (37)$$

or

$$\langle g_i | S | g_i \rangle \xrightarrow{(13)} -\langle g_i | d_i \rangle = 2(f_i - f_0) \quad (38)$$

Si l'on fait l'hypothèse que la fonction décroît linéairement au long de $|d_i\rangle$ on pourra utiliser l'équation (37) corrigée comme voici :

$$\langle g_i | \lambda d_i \rangle = -2(f_i - f_0) \quad (39)$$

d'où

$$\lambda = 2 \frac{f_0 - f_i}{\langle g_i | d_i \rangle} \quad (40)$$

ou mieux encore

$$\lambda = 2 \frac{|f_0 - f_i|}{\langle g_i | d_i \rangle} \quad (41)$$

Finalement le critère logique de choix parmi les deux prédictions de λ est le suivant :

$$\lambda_i = \min \left(\frac{\text{pas}}{\text{long} |d_i\rangle} ; \frac{2 |f_0 - f_i|}{\langle g_i | d_i \rangle} \right) \quad (42)$$

Ce critère est d'une efficacité surprenante.

10.2 - On ne connaît rien sur la valeur du minimum de la fonction.

On prend comme intervalle initial pour l'itération i la longueur du vecteur $|\sigma_{i-1}\rangle$ trouvé au cours de l'itération $(i-1)$. Quoique pas toujours, $\text{long } |\sigma_{i-1}\rangle$ est grande loin du minimum et diminue dans les proximités

$$\lambda_i = \min \left(\frac{\text{pas}}{\text{long } |d_i\rangle}, \frac{\text{long } |\sigma_{i-1}\rangle}{\text{long } |d_i\rangle} \right) \quad (43)$$

11 - Tests de convergence

Un sous-programme est un sous-système qui, lancé à un certain moment par le système gérant, exécute une séquence de tâches et une fois cette dernière finie retourne le contrôle au système d'appel.

Normalement, il n'y a pas de difficulté pour fixer la fin de la séquence. Si l'on regarde le problème de plus près, l'on s'aperçoit qu'un sous-programme représente la description formelle d'une trajectoire depuis un état initial d'un calculateur, en passant par un nombre probablement très élevé d'états intermédiaires, à un état final ou de sortie. L'ensemble des états de sortie doit être spécifié a priori, de façon à ce que, quel que soit l'état initial, on arrive à l'un de ceux de l'ensemble de sortie, après un temps fini. En pratique on a l'habitude de fixer cet ensemble par un ensemble d'équivalences vis à vis de certaines valeurs des

variables du problème, lorsqu'il est connu que de telles valeurs sont atteignables. Or, lorsqu'il n'en est pas ainsi, le sous-programme est un système incontrôlable.

C'est cela précisément qui peut arriver dans un sous-programme de minimisation, où la seule chose que l'on sait est que la condition théorique exacte nécessaire pour avoir un minimum local est :

$$|g(x_{\min})\rangle = |0\rangle$$

qui sont des valeurs pratiquement non atteignables .

On a retenu le test suivant :

Si $|d_i\rangle_K \leq E$ pour $K = 1, 2, \dots, n$

et si $|\hat{g}_i\rangle_K \leq E$ pour $K = 1, 2, \dots, n$

la convergence est assurée et l'on retourne le contrôle au programme d'appel.

Mais comme d'après ce que l'on vient de dire, cela n'est peut-être pas suffisant , on fixe comme état de sortie quelque chose de sûrement atteignable, un nombre maximum d'itérations (NITER) à fournir par l'utilisateur.

Il faut aussi prévenir la division par une quantité très petite, ce qui dépasse la capacité d'une mémoire du calculateur et peut produire une interruption de programme. A la rigueur, il faudrait prendre ces mêmes précautions dans les formules d'interpolation de MINLIC, qui pourraient elles aussi interrompre le programme.

IV - LISTE DE PROGRAMMES ET RESULTATS

Le sous-programme DAVMIN est structuré de façon à n'avoir à changer que la carte d'appel (CALL MINLIC ou CALL MINLIF) lorsque l'on utilise l'une ou l'autre M.M.U. Ceci veut dire évidemment qu'il existe des instructions qui, nécessaires pour une méthode, sont superflues pour l'autre.

Il faut ajouter aussi que, dans certains cas, nous avons obtenu des meilleurs résultats (au point de vue nombre de coups) avec des programmes plus simples (sans les modifications qu'on a apporté par la suite). Mais dans d'autres cas, ces modifications et peut être d'autre que vous seriez en mesure d'introduire, se sont avérées absolument nécessaires.

Machine I.B.M. 360-44 du C.E.R.A.

Fonctions-test

a) Rosembrock à 2 dimensions :

$$Y = 100 (X_2 - X_1^2)^2 + (-1 - X_1)^2$$

dont

$$X_{\text{opt}} = (-1; 1)$$

$$Y_{\text{opt}} = 0$$

Voici l'allure des résultats obtenus pour :
point initial $X_0 = (-1,2 ; 1)$

fest = 0, min = 1, E = 10^{-6} , pas = 1.

DAVMIN + MINLIC

Itéra- tion	Coups	valeur	X_1	X_2
1	3	4.1281003952	-1.0301990509	1.0693054199
2	5	3.8591585159	-0.9173573256	0.7987771630
3	7	2.9381237030	-0.6404709816	0.3605058789
4	10	0.7661653757	0.1322034597	0.0060346127
5	12	0.7603486180	0.1287385821	0.0130349398
6	15	0.5912057161	0.2733167410	0.0495749116
7	17	0.3368033767	0.4951993227	0.2165902257
8	20	0.3209164739	0.4508219361	0.1893407702
9	23	0.2808499932	0.5291358232	0.2556666136
10	25	0.2323293686	0.5185519457	0.2665784359
11	27	0.1762835383	0.6959520578	0.4553943872
12	29	0.1453080773	0.6330236197	0.3904055357
13	32	0.1190776229	0.7015418410	0.4748402834
14	35	0.0159710869	0.9057968259	0.8120436072
15	37	0.0110020339	0.8961278200	0.8015869260
16	39	0.0026144218	0.9614238143	0.9209796786
17	41	0.0010720959	0.9917566180	0.9804123640
18	43	0.0000628926	0.9920699000	0.9842104912
19	45	0.0000004460	0.9998858571	0.9997059107
20	47	0.0000000026	0.9999491572	0.9998987913
21	62	0.0000000000	0.9999999404	0.9999999404

avec un programme antérieur (appelons-le (DAVMIN+MINLIC)₁)

20 48 0.0. 1.0000000000 1.0000000000

Pour ce même cas on a supposé inconnu le minimum :

$$\min = 2$$

On a trouvé :

Itération 25, coups 102, minimum 0.0 et
point (1.0000000000 ; 1.0000000000)

DAVMIN + MINLIF

Données de départ :

$$X_0 = (1,2 ; 1)$$

fest = 0, min = 1, E = 10^{-6} , PRE = 0,005, pas = 1

Résultats :

Itéra- tion	Coups	Valeur	X_1	X_2
1	10	4.1295461655	-1.0285615921	1.0699739456
2	22	4.1235618591	-1.0300559998	1.0659503937
3	34	3.4610309601	-0.8124718666	0.6181606650
4	46	3.1711282730	-0.6728329062	0.3916499615
5	70	1.0283098221	-0.0001638532	-0.0167284012
6	82	1.0147018433	-0.0058521107	-0.0054109357
7	94	0.7602392435	0.1707445383	0.0022140071
8	106	0.6698098779	0.2697935104	0.0358279571
9	115	0.4808720350	0.3094622493	0.1021148562
10	127	0.3475463986	0.4369973540	0.1734811068
11	139	0.3142336901	0.5025179386	0.2267104387
12	161	0.0501866192	0.7935883403	0.6210756302
13	168	0.0456530326	0.7870128751	0.6176876426
14	178	0.0281912722	0.8561884165	0.7243928313
15	196	0.0046143457	1.0544624329	1.1159505844
16	203	0.0038780414	1.0600223541	1.1253061295
17	211	0.0009543635	1.0245161057	1.0477533340
18	224	0.0000135691	0.9983247519	0.9963242412
19	262	0.0000000388	0.9998216033	0.9996515512
20	291	0.0	1.0000000000	1.0000000000

Un programme antérieur (DAVMIN + MINLIF₁) nous avait fourni
comme résultat pour PRE = 0.005 :

Itération 26, coups 219, minimum 0.0 et
point (1.0000000000 ; 1.0000000000)

b) Fonction à 3 dimensions :

$$Y = 100(X_2 - X_1^2)^2 + (X_1 - 1)^2 + (X_3 - 1)^2$$

dont : $X_{\text{opt}} = (1; 1; 1)$

$$Y_{\text{opt}} = 0$$

Données de départ :

$$X_0 = (-2; 1; 1) \text{ fest} = 0, \text{ min} = 1, E = 10^{-6}$$

pas = 1

DAVMIN+MINLIC

Itération 242, coups 519, minimum 0.0 et
point (1.0000000000 ; 1.0000000000 ; 1.0000000000)

(DAVMIN + MINLIC) avait donné :

Itération (32), coups (89) et le reste pareil.

DAVMIN + MINLIF

Itération (26), coups (258), minimum (0.0000000000) et
 $X_{\text{min}} = (0.9999996424 ; 0.9999992847 ; 1.0000000000)$.

(DAVMIN + MINLIF)₁ :

Itération (28), coups (270), minimum (0.0) et $X_{\text{min}} = (1.0000000000 ; 1.0000000000 ; 1.0000000000)$

C₁) Fonctions à 4 dimensions

$$Y = (X_1^2 + 2X_2^2 + 3X_3^2 + 4X_4^2)^2$$

dont :

$$X_{\text{opt}} = (0 ; 0 ; 0 ; 0)$$

$$Y_{\text{opt}} = 0$$

Données de départ :

$$X_0 = (1;1;1;1), \text{ fest} = 0, \text{ min} = 1, E = 10^{-6}, \text{ pas} = 1$$

DAVMIN + MINLIC :

Itération (27), coups (65), $\psi = 0.0000000003$

$X = (-0.0033094874; -0.0002021121; 0.0015122739; -0.0001353810)$

DAVMIN+MINLIF :

Itération (19), coups (380), $\psi = 0.0000000000$

$X = (-0.0001791904; -0.0005368988; -0.0000984832; -0.0000235401)$

$$C_2) \quad Y = \sum_{K=1}^4 K^2 X_K^2$$

dont $X_{\text{opt}} = (0; 0; 0; 0)$

$$Y_{\text{opt}} = 0$$

Données de départ :

$X_0 = (1; 1; 1; 1)$, fest=0; min = 1, $E = 10^{-6}$, pas = 1

DAVMIN+MINLIC :

Itération (5), coups (12), min = (0.0000000000)

$X_{\min} = (0.0000000034; 0.0000000255; 0.0000001591; -0.00000001832)$

DAVMIN + MINLIF :

Itération (9) ; coups (141), min = (0.0000000000)

$X_{\min} = (0.0000000023; 0.0000000621; 0.0000000297; 0.0000001797)$

$$c_3) \quad Y = (X_1 + 10 X_2)^2 + 5(X_3 - X_4)^2 + (X_2 - 2 X_3)^4 + 10(X_1 - X_4)^4$$

$$\text{dont} \quad X_{\text{opt}} = (0; 0; 0; 0)$$

$$Y_{\text{opt}} = 0$$

Données de départ :

$X_0 = (3; -1; 0; 1)$, fest = 0, min = 1, $E = 10^{-6}$, pas = 1

DAVMIN+ MINLIC :

Itération (41), coups (132), = 0.0000000000

$X = (-0.0019750418; 0.0001977362; -0.0010074954; -0.0010073620)$

DAVMIN + MINLIF :

Itération (35), coups (576), = 0.0000000000

X =(0.0000112972;-0.0000011294;-0.0001969963;-0.0001970251)

d) Fonction à 6 dimensions

$$Y = 100(X_2 - X_1^2)^2 + (X_1 - 1)^2 + 2[100(X_4 - X_3^2)^2 + (X_3 - 1)^2] + 3[100(X_6 - X_5^2)^2 + (X_5 - 1)^2]$$

dont :

$$X_{\text{opt}} = (1; 1; 1; 1; 1; 1)$$

$$y_{\text{opt}} = 0$$

Données de départ :

$X_0 = (0; 0; 0; 0; 0; 0)$, fest = 0, min = 1, E = 10^{-6} , pas = 1

DAVMIN + MINLIC :

Itération (49), coups (129), minimum = 0.0000000001

$X_{\text{min}} = (0.9999929070 ; 0.9999863505 ; 1.0000028610 ; -1.0000057220 ;$
0.9999999404 ; 0.9999998212)

DAVMIN + MINLIF :

Itération (40), coups (463), minimum = 0.0

$X_{\min} = (1.0000000000 ; 1.0000000000 ; 1.0000000000 ; 1.0000000000 ;$
 $1.0000000000 ; 1.0000000000)$

(DAVMIN + MINLIF) donnait

Itération (250), coups (1111), $y = 0.00000000$

$X = (1.00000477 ; 1.00000954 ; 1.00000381 ; 1.00000858 ;$
 $0.99999976 ; 0.99999911)$

Toutes ces fonctions ont été prises dans les articles cités en référence, dans la thèse de Recio et dans un rapport au C.E.R.A. intitulé "La Méthode du Modèle).

V - REMARQUES FINALES

12 - Bibliographie sur la méthode de Davidon et sur une autre méthode de minimisation qui ne calcule pas des dérivées.

Fletcher et Reeves [3] ont proposé une méthode de gradients conjugués qui ne demande pas le stockage et la manipulation d'une matrice et utilise en particulier le même algorithme de minimisation linéaire proposé en [2].

Stewart [8] propose une méthode pour calculer le gradient par différences ce qui le permet de profiter des avantages de Davidon sans avoir à calculer des dérivées.

Quant aux méthodes qui, déjà à l'origine, n'ont pas à calculer des dérivées, nous avons fixé notre attention sur celle proposée par POWELL [9], méthode de convergence quadratique basée sur une recherche de directions conjuguées par rapport à la matrice symétrique de dérivées secondes au voisinage du minimum. Nous avons fait une première étude et programmation de cette méthode, en utilisant un sous-programme basé sur Fibonacci (1) et les résultats ont été assez encourageants. Pour la fonction de Rosenbrock à 2 dimensions (figure 9), avec $X_0 = (-1.2 ; 1)$ l'itération n° 11 nous donnait déjà $X_1 = 1.00065136$, $X_2 = 1.00125694$, valeur de la fonction $f = 0.00000063$ après 488 coups (mais sans calcul du gradient). Ceci avec un programme qui n'était pas élaboré. La méthode repose sur une minimisation exacte le long de chaque direction pour trouver des nouvelles directions qui soient à la fois conjuguées et linéairement indépendantes. La théorie n'est pas tout à fait mise au point comme d'ailleurs vient de signaler Zangwill [10]. Pour une étude comparative de cette méthode avec deux autres méthodes directes, voir Fletcher [11].

Pour une très abondante bibliographie sur les méthodes d'optimisation statique voir Lavi [12].

(1) Nous avons trouvé spécialement peu claire la section 8 de l'article de Powell qui présente une méthode de minimisation sur une direction qui ne consiste en fait qu'à faire passer une parabole $y = AX^2 + Bx + C$ par trois points alignés. Les formules sont exactes et la propriété qu'il utilise pour prédire la dérivée seconde est basée sur le fait suivant : lorsqu'on connaît A il ne faut que deux points pour déterminer B et C.

13 - Application de la méthode à l'optimisation paramétrique de systèmes dynamiques.

Soit un système comme celui décrit à la figure 19.

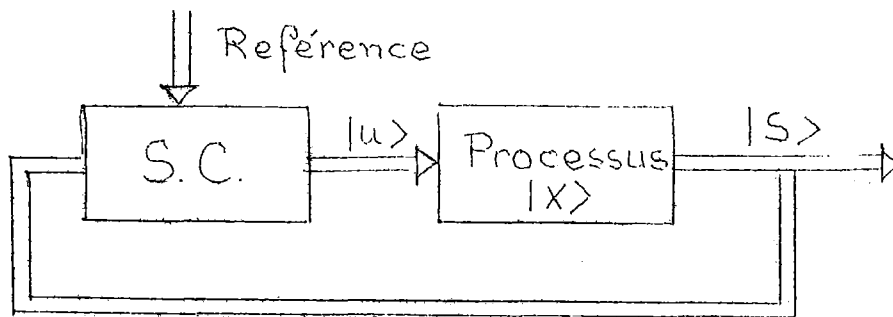


Figure 19

L'optimisation paramétrique consiste à choisir une structure du système de commande et chercher à optimiser le fonctionnement (un critère) du processus en modifiant en boucle fermée les paramètres du S.C. (voir Boudarel et al. [13]).

Caballero [14] a utilisé avec succès la méthode de Davidon pour résoudre le problème suivant :

Etant donné le processus décrit par l'équation d'état

$$\begin{aligned} |\dot{X}\rangle &= A |X\rangle + B |u\rangle \end{aligned} \quad \left(\begin{array}{l} |X\rangle \text{ vecteur } (4 \times 1) \\ |u\rangle \text{ vecteur } (2 \times 1) \end{array} \right)$$

transférer le système depuis $|X(0)\rangle = |0\rangle$ à $|X(T)\rangle = |X_F\rangle$
 en minimisant le critère :

$$\text{Critère} = T + \frac{K}{2} \langle \varepsilon | \varepsilon \rangle$$

avec K coefficient de pénalisation, $|\varepsilon\rangle = |X(T)\rangle - |X_F\rangle$
 écart final entre l'état obtenu et l'état désiré.

Les commandes sont assujetties à des contraintes de saturation.

$$|u| \leq M$$

Gigon dans sa thèse [15] envisage le problème suivant ; soit
 un système (figure 20) :

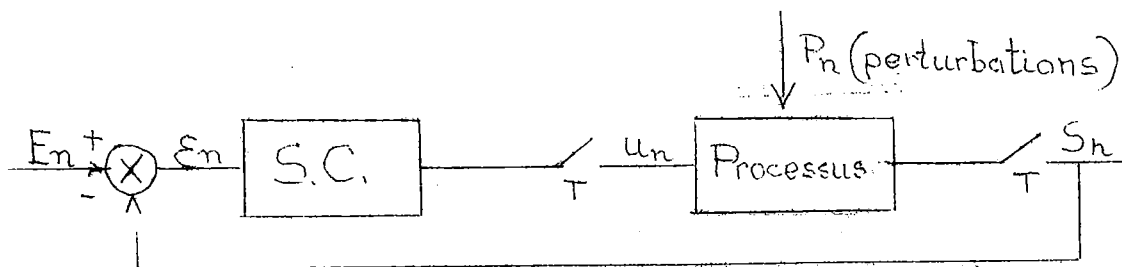


Figure 20

$$\text{Processus discrétisé} \begin{cases} |X_{n+1}\rangle = A |X_n\rangle + |B\rangle u_n \\ s_n = \langle C | X_n \rangle \end{cases}$$

Le correcteur est supposé à structure linéaire de fonction de transfert en Z :

$$Z(S.C.) = \frac{a_1 + a_2 Z^{-1} + \dots + a_K Z^{-(K-1)}}{1 + b_1 Z^{-1} + \dots + b_m Z^{-m}}$$

Et il s'agit de trouver la fonction Z qui minimise le critère :

$$\text{Critère} = \sum_{n=1}^N (\epsilon_n^2 + K u_n^2)$$

en tenant compte des perturbations et des saturations sur u_n .

Gigon, en fait, compare quatre méthodes de programmation non linéaire : Gradient, Paxtan, Gauss-Newton et Davidon. Il nous a communiqué un des résultats préliminaires qu'il a trouvé :

$$F.T.Processus = \frac{K}{p^2 a^2}$$

<u>Méthode</u>	<u>Temps de calcul</u>	<u>Précision</u>	<u>Structure</u>
Gradient	1 h 15	2 %	à 5 paramètres
Paxtan-Fibonacci	44 m.	$\sim 10^{-4}$	"
Gauss-Newton	56 s.	$\sim 10^{-6}$	"
Davidon-Davidon	54 s.	$\sim 10^{-5}$	"

```

C.
C   PROGRAMME DE TEST DE DAVMIN
C   *****
  DIMENSION XO(10),GRO(10),D(10),S(10,10),X1(10),SIGM(10),GR1(10),
  1Y(10)
  1 READ(5,1000)N,MIN,IECRI,NITER
    READ(5,1001)(XO(I),I=1,N)
    READ(5,1002)((S(I,J),J=1,N),I=1,N)
    READ(5,1003)PAS,E
    READ(5,1004)FEST
1000 FORMAT(3(I2),I5)
1001 FORMAT(5F12.7)
1002 FORMAT(14F5.3)
1003 FORMAT(2F12.7)
1004 FORMAT(F8.4)
    CALL DAVMIN(N,XO,FEST,E,S,MIN,PAS,NITER,IECRI,FO)
    GO TO 1
    STOP
    END

```

```

C.
C SUBROUTINE DAVMIN(N,XO,FEST,E,S,MIN,PAS,NITER,IECRI,FO)
C OBJET %MINIMISATION D'UNE FONCTION DIFFERENTIABLE DANS UN ESPACE
C A N DIMENSIONS SANS CONTRAINTES (BASEE SUR DAVIDON)
C DEFINITION DES ARGUMENTS %
C N NOMBRE DE VARIABLES
C XO A L'ENTREE VECTEUR ESTIME POUR INITIALISER
C PAUSE
C A LA SORTIE VECTEUR MINIMUM
C FEST VALEUR ESTIMEE DU MINIMUM DE LA FONCTION
C E PRECISION AFFICHEE
C S MATRICE DE TRANSFORMATION DU VECTEUR DIRECTION
C A L'ENTREE, MATRICE QUELCONQUE SYMETRIQUE STRICT POSITIVE
C A LA SORTIE MATRICE ESTIMEE INVERSE DE L'HESSIAN
C MIN =1,SI L'ON CONNAIT FEST
C =2,SI L'ON NE CONNAIT PAS
C PAS INTERVALLE DONNE PAR L'USAGER POUR LA RECHERCHE SUR UNE DIRECTION
C NITER NOMBRE MAXIMUM D'ITERATIONS PERMIS PAR L'USAGER
C IECRI PARAMETRE QUI CONTROLE LE DEBIT D'INFORMATION ECRITE
C FO VALEUR TROUVEE
C REFERENCES EXTERNES %
C SUBROUTINE GRAD(X,G).ENTREE X VECTEUR POINT.SORTIE G VECTEUR GRADIENT
C FUNCTION VALEUR(X).ENTREE X VECTEUR POINT.
C SUBROUTINE MINLIC OU MINLIF
C AUTRES VARIABLES IMPORTANTES %
C ITER COMPTEUR D'ITERATIONS
C NC COMPTEUR DU NOMBRE DE FOIS QU'ON CALCULE LA FONCTION
C D(I) VECTEUR DIRECTION, AUSSI UN PRODUIT VECTORIEL
C DERIVO DERIVEE DE LA FONCTION AU POINT XO
C DELTA INTERVALLE POUR COMPARAISON
C NCF COMPTEUR A L'INTERIEUR DE MINLIC OU MINLIF
C
C SIGM(I) VECTEUR DEPLACEMENT AU COURS DE CHAQUE ITERATION
C X1(I) MEILLEUR POINT TROUVE AU COURS DE CHAQUE ITERATION
C Y(I) VECTEUR INCREMENT DU GRADIENT
C *****
C SUBROUTINE DAVMIN(N,XO,FEST,E,S,MIN,PAS,NITER,IECRI,FO)
C DIMENSION XO(10),GRG(10),D(10),S(10,10),X1(10),SIGM(10),GR1(10),
C 1Y(10)
C KK=0
C ITER=0
C NC=0
C WRITE(6,901)
C 901 FORMAT(' ITERATION',2X,'COUPS',6X,'VALEUR',9X,'X1',12X,'X2',12X,
C 1'X3',12X,'X4',12X,'X5',12X,'X6'/)
C DELTA=PAS
C I=IECRI
C 100 DEBUT D'UNE ITERATION
C ITER=ITER+1
C IF(ITER-1)201,200,201

```


C CALCUL VECTEUR DIRECTION ET LONGUEUR

```
200 CALL GRAD(X0,GRO)
    FO=VALEUR(X0)
    NC=NC+1
201 DO 202 I=1,N
    D(I)=0.
    DO 202 J=1,N

202 D(I)=D(I)-S(I,J)*GRO(J)
    PRO=0.
    DO 203 I=1,N
203 PRO=PRO+D(I)*D(I)
    PRO=SQRT(PRO)
    IF(PRO-1.0E-73)600,600,300
```

C DETERMINATION DE L'INTERVALLE INITIAL ET MINIMISATION LINEAIRE

```
300 DERIVO=0.
    DO 301 I=1,N
301 DERIVO=DERIVO+GRO(I)*D(I)
    IF(DERIVO)308,309,309
C      REINITIALISATION %MATRICE UNITE
309 DO 310 I=1,N
    DO 310 J=1,N
310 S(I,J)=0.
    DO 311 I=1,N
311 S(I,I)=1.
    GO TO 201
308 IF(ABS(DERIVO)-1.0E-73)600,600,302
302 GO TO (303,305),MIN
303 DELTA=2.*ABS((FO-FEST)/DERIVO)
    BOUND=PAS/PRO
    IF(DELTA-BOUND)306,304,304
304 DELTA=BOUND
    GO TO 306

305 DELTA=DELTA/PRO
```

C SI L'ON UTILISE SUBROUTINE MINLIC CHANGER LA CARTE ETIQUETTEE 306
C POUR CELLE-CI %306 CALL MINLIC(N,X0,D,E,NCF,FO,X1,DELTA,DERIVO,MOTIF)

```
306 CALL MINLIF(N,X0,D,NCF,X1,FO,DELTA,PRO,E,MOTIF)
    NC=NCF+NC
    IF(ITER-II)400,307,400
```

307 II=II+IECRI

C CARTES D'ECRITURE A CHANGER EN FONCTION DE L'INFORMATION DESIREE
WRITE(6,902)ITER,NC,FO,(X1(I),I=1,N)

902 FORMAT(2X,I3,2X,I5,2X,7(F13.10,2X))

C CALCUL DU VECTEUR DEPLACEMENT

```
400 DO 401 I=1,N
401 SIGM(I)=X1(I)-X0(I)
    GO TO (405,402),MIN
```

C DETERMINATION DU NOUVEAU INTERVALLE AU CAS OU FEST N'EST PAS CONNU
402 DELTA=0.

```

      DO 403 I=1,N
403  DELTA=DELTA+SIGM(I)*SIGM(I)
      DELTA=SQRT(DELTA)
      IF(DELTA-PAS)405,405,404
404  DELTA=PAS
405  IF(MOTIF-1)407,406,407
406  DELTA=10.*E
407  DO 408 I=1,N
408  X0(I)=X1(I)

C      TEST DE CONVERGENCE
500  IF(ITER-NITER)501,600,600

501  DO 502 I=1,N
      IF(ABS(D(I))-E)502,502,700
502  CONTINUE
      DO 503 I=1,N
      IF(ABS(SIGM(I))-E)503,503,700
503  CONTINUE
600  RETURN
700  CALL GRAD(X1,GR1)
      IF(MOTIF-1)709,703,709
703  IF(KK-1)704,705,704
704  KK=KK+1
      GO TO 706
705  WRITE(6,903)
903  FORMAT(' CONVERGENCE NON ASSUREE,CHANGER A UN POINT INITIAL PROCHE
1A CELUI-CI,VARIER L'INTERVALLE OU RETOURNER AU PROGRAMME D'APPEL')
      GO TO 600
C      REINITIALISATION DU SYSTEME. MATRICE %UNITE.DIRECTION %
C      GRADIENT NEGATIF
706  DO 707 I=1,N
      DO 707 J=1,N
707  S(I,J)=0.

      DO 708 I=1,N

708  S(I,I)=1.

      GO TO 716
709  DO 701 I=1,N
701  Y(I)=GR1(I)-GR0(I)
      PRO=0.
      DO 702 I=1,N
702  PRO=PRO+SIGM(I)*Y(I)
C      FORMATION DE LA NOUVELLE MATRICE S
      DO 710 I=1,N
      D(I)=0.
      DO 710 J=1,N
710  D(I)=D(I)+S(I,J)*Y(J)
      PRO1=0.
      DO 711 I=1,N
711  PRO1=PRO1-Y(I)*D(I)

```

```

      IF (ABS (PRO) - 1.0E-73) 600, 600, 712
712 IF (ABS (PRO1) - 1.0E-73) 600, 600, 713
713 DO 714 I=1, N
      DO 714 J=1, N
714 S (I, J) = S (I, J) + SIGM (I) * SIGM (J) / PRO + D (I) * D (J) / PRO1
      DO 715 I=1, N
715 XO (I) = X1 (I)
716 DO 717 I=1, N
717 GRO (I) = GR1 (I)
      IF (MOT IF - 1) 718, 719, 718
718 KK = 0

719 MOTIF = 0
800 GO TO 100

      END

```

```

C.
C   SUBROUTINE MINLIF(N,XIN,D,NCF,X1,YIN,AINT,ALONG,E,MOTIF)
C   OBJET %MINIMISATION LE LONG D'UNE DIRECTION (BASEE SUR FIBONACCI)
C   DEFINITIONS DES ARGUMENTS %
C   N       NOMBRE DE VARIABLES
C   XIN     POINT INITIAL DE LA RECHERCHE
C   D       VECTEUR DIRECTION
C   NCF     COMPTEUR DES FOIS QU'ON APPELLE VALEUR
C   X1      A LA SORTIE MEILLEUR POINT TROUVE
C   YIN     A LA SORTIE VALEUR DE LA FONCTION AU POINT X1
C   AINT    PROJECTION DE L'INTERVALLE SUR LE VECTEUR D(I)
C   ALONG   LONGUEUR DE D(I)
C   E       PRECISION VOULUE
C   MOTIF =1 RETOURNE LE CONTRÔLE A DAVAIN LORSQUE AINT TROP PETIT
C   REFERENCES EXTERNES
C   FUNCTION VALEUR
C   AUTRES VARIABLES IMPORTANTES %
C   PRE     PRECISION DANS LA RECHERCHE
C   F0,F1,F2 TERMES SERIE FIBONACCI
C   Y1,Y2 VALEURS DE LA FONCTION
C   R0,R1,R2 SOUS INTERVALLES AVANT PROJECTION SUR D(I)

C   *****

C   SUBROUTINE MINLIF(N,XIN,D,NCF,X1,YIN,AINT,ALONG,E,MOTIF)
C   DIMENSION XIN(10),D(10),X2(10),X1(10)
C   NCF=0
C   NORMALISATION DU VECTEUR DIRECTION
C   DO 1 I=1,N
1  D(I)=D(I)/ALONG
  AINT=AINTE*ALONG
3  PRE=0.005
  IF(AINT-5.*PRE)6,6,10
6  IF(AINT-E)7,9,9
7  MOTIF=1
  DO 8 I=1,N
8  X1(I)=XIN(I)
  GO TO 34
9  PRE=E
C   OBTENTION TERMES SERIE FIBONACCI
10 F0=1.
  F1=1.
11 F2=F0+F1
  IF(AINT-PRE*F2)13,13,12
12 F0=F1

  F1=F2

  GO TO 11
13 R0=0.
  DELTA=AINTE/F2

  R1=DELTA*F0
  R2=DELTA*F1

```

```

DO 14 I=1,N
  X1(I)=XIN(I)+R1*D(I)
14 X2(I)=XIN(I)+R2*D(I)
  Y1=VALEUR(X1)
  Y2=VALEUR(X2)
  NCF=NCF+2
C   REBOURS SERIE FIBONACCI
15 F2=F1
  F1=F0
  F0=F2-F1
  IF(Y1-Y2)19,21,21
C   REDUCTION DE L'INTERVALLE D'INCERTITUDE
19 R2=R1
  Y2=Y1
  R1=R0+DELTA*F0
  DO 20 I=1,N
    X2(I)=X1(I)
20 X1(I)=XIN(I)+R1*D(I)
  Y1=VALEUR(X1)
  NCF=NCF+1
  GO TO 23
C   REDUCTION DE L'INTERVALLE D'INCERTITUDE
21 R0=R1

  R1=R2
  Y1=Y2
  R2=R0+DELTA*F1
  DO 22 I=1,N
    X1(I)=X2(I)

22 X2(I)=XIN(I)+R2*D(I)
  Y2=VALEUR(X2)
  NCF=NCF+1
C   EST-CE QUE LA RECHERCHE EST FINIE
23 IF(F2-2.)24,24,15
24 IF(Y1-Y2)29,25,25
C   VERIFIER SI LE POINT TROUVE EST MEILLEUR QUE LE POINT INITIAL
25 IF(Y2-YIN)27,7,7
27 YIN=Y2
  DO 28 I=1,N
28 X1(I)=X2(I)
  SIGMA=R2
  GO TO 31
29 IF(Y1-YIN)30,7,7
30 YIN=Y1
  SIGMA=R1
31 IF(AINT-3.*E)34,34,36
36 IF(AINT-SIGMA-2.*PRE)32,32,34
C   DEPLACEMENT DE L'INTERVALLE POUR NOUVELLE RECHERCHE
32 DO 33 I=1,N

```

```
33 XIN(I)=X1(I)
    AINT=2.*AINT
    GO TO 3
C    DENORMALISATION DU VECTEUR DE DIRECTION
34 DO 35 I=1,N
35 D(I)=D(I)*ALONG
    RETURN
    END
```

SUBROUTINE MINLIC(N,XO,D,E,NCF,FO,X,DELTA,DERIVO,MOTIF)
 OBJET %MINIMISATION LE LONG D'UNE DIRECTION(BASEE SUR UNE METHODE
 D'INTERPOLATION CUBIQUE DE DAVIDON)

.....
 DEFINITION DES ARGUMENTS %
 N NOMBRE DES VARIABLES
 XO POINT INITIAL DE RECHERCHE
 D VECTEUR DIRECTION
 E PRECISION
 NCF COMPTEURS DES FOIS QU'ON APPELLE VALEUR
 FO A L'ENTREE VALEUR DE LA FONCTION AU POINT XO
 A LA SORTIE VALEUR AU MEILLEUR POINT
 X A LA SORTIE MEILLEUR POINT TROUVE.,AUSSI POINT COURANT DANS LA
 RECHERCHE
 DELTA INTERVALLE PROJETE.,AUSSI SOUS INTERVALLE AU COURS RECHERCHE
 DERIVO DERIVEE DE LA FONCTION AU POINT XO
 MOTIF =1 RETOURNE LE CONTROLE A DAVMIN PAR MESURE DE SECURITE
 REFERENCES EXTERNES %
 SUBROUTINE GRAD(X,G)
 FUNCTION VALEUR(X)
 AUTRES VARIABLES IMPORTANTES %
 R INTERVALLE DE RECHERCHE
 IK,IKK PARAMETRES QUI PERMETTENT DE DETECTER CERTAINES UNIMODALITES

DERIV1 DERIVEE AU POINT X1
 DERIV2 DERIVEE AU POINT X2
 W,Z VARIABLES UTILISEES POUR L'INTERPOLATION CUBIQUE

SUBROUTINE MINLIC(N,XO,D,E,NCF,FO,X,DELTA,DERIVO,MOTIF)
 DIMENSION X(10),D(10),GR(10),XO(10)

CHANGE

NCF=0
 A=1.
 R=DELTA
 1 DELTA=0.
 DO 2 I=1,N
 2 X(I)=XO(I)
 3 DO 4 I=1,N
 4 X'(I)=X(I)+R*D(I)
 F1=VALEUR(X)
 NCF=NCF+1
 CALL GRAD(X,GR)
 DERIV1=0.
 DO 5 I=1,N
 5 DERIV1=DERIV1+GR(I)*D(I)
 IF(DERIV1)6,10,10
 6 IF(F1-FO)7,7,9
 7 DO 8 I=1,N
 8 XO(I)=X(I)
 DELTA=R+DELTA
 R=DELTA
 FO=F1
 DERIVO=DERIV1

```

      GO TO 3
C    REDUCTION DU PAS EN VUE D'UNE POSSIBLE ERREUR
      9 A=10.*A
      R=R/A
      GO TO 23
    10 T=0.
    11 IK=0
      IKK=0
      Z=3.*(F0-F1)/R+DERIVO+DERIV1
      W=SQRT(Z**2-DERIVO*DERIV1)
      DELTA=R*(DERIV1+W-Z)/(DERIV1-DERIVO+2.*W)
      DO 12 I=1,N
    12 X(I)=X(I)+(T-DELTA)*D(I)
      F2=VALEUR(X)
      NCF=NCF+1
      CALL GRAD(X,GR)
      IF(F2-F0)13,13,15
    13 IKK=IKK+1
      IF(F2-F1)26,26,14
    14 IKK=IKK+1
      GO TO 16
    15 IK=IK+1

    16 DERIV2=0.
      DO 17 I=1,N
    17 DERIV2=DERIV2+GR(I)*D(I)
      IF(DERIV2)20,18,18
    18 IKK=IKK+1
      IF(IKK-3)19,22,22
    19 F1=F2
      DERIV1=DERIV2
      R=R-DELTA
      GO TO 10
    20 IK=IK+1
      IF(IK-2)21,22,22
    21 F0=F2
      DERIVO=DERIV2
      R=DELTA
      T=R
      GO TO 11
C    REDUCTION DU PAS EN VUE D'UNE POSSIBLE ERREUR
    22 A=10.
      R=(R-DELTA)/A
    23 IF(R-E)24,24,1
    24 DO 25 I=1,N
    25 X(I)=X0(I)
      MOTIF=1
      GO TO 27
    26 F0=F2
    27 RETURN

```


VI - REFERENCES

- [1] - Davidon W.C.
"Variable Metric Method for minimization" A.E.C. Research
and Development Report, ANL-5990 (Rev) 1959.
- [2] - Fletcher R. and Powell
"A rapidly convergent descent method for minimization"
Computer Journal, 6 (1963)
- [3] - Fletcher R. and Reeves
"Function minimization by conjugate gradients"
Computer Journal, 7 (1964)
- [4] - Wilde D.J.
"Méthodes de Recherche d'un optimum"
Dunod 1966 - Traduction française de "Optimum seeking
methods" Prentice Hall 1964.
- [5] - Recio R.
"Application de la méthode de Fibonacci aux problèmes
de programmation non-linéaire"
Thèse de l'Ecole Nationale Supérieure de l'Aéronautique
(1967).
- [6] - Boudarel R., Delmas J. Guichet
"Application de la programmation non-linéaire à la
commande optimale des processus"
fascicule 1 au C.E.R.A. pp. 28-29 (1965)

- Il n'existe pas de référence 7

- [8] - Stewart G.W.
"A modification of Davidon's minimization Method to accept difference approximations of derivatives"
Journal of the Association for Computing Machinery
Volume 14 - N° 1 - (1967)
- [9] - Powell M.J.D.
"An efficient method for finding the minimum of a function of several variables without calculating derivatives" - Computer Journal, 7 (1964)
- [10] - Zangwill W.I.
"Minimizing a function without calculating derivatives"
Computer J., 10 (1967)
- [11] - Fletcher R.
"Function minimization without evaluating derivatives, a review" - Computer J. 8 (1965)
- [12] - Lavi A., Vogl T.P.
"Recent advances in optimization techniques"
J. Wiley (1966)
- [13] - Boudarel R., Delmas J., Guichet
"Application de la programmation non-linéaire à la commande optimale des processus" - Fascicule 2 au C.E.R.A. - pp. 94-104

[14] - Caballero P.A.

"Optimisation paramétrique des systèmes linéaires
multivariables en temps minimum" -

Thèse à l'Ecole Nationale Supérieure de l'Aéronautique
Décembre 1967.

[15] - Gigon D.

"Optimisation paramétrique des systèmes de commande
à données numériques" - Thèse en cours pour un
Doctorat d'Université.